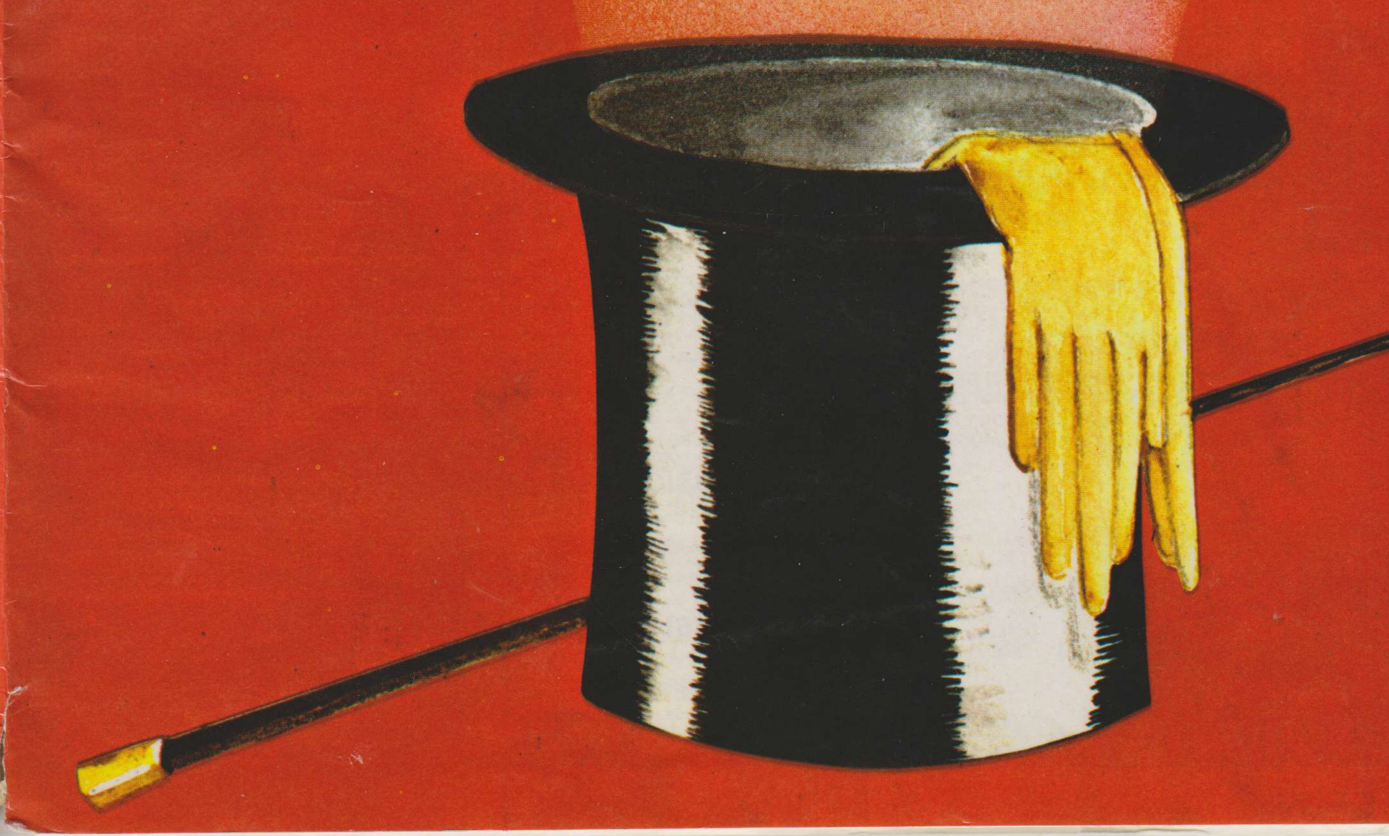
Commodore WORLD

Especial nº 1

100 programas



¡Hazte socio del Club de Soft de España!

Tendrás multitud de ventajas...

— Recibirás puntual información de las novedades de soft que haya en el mercado.
Te enviaremos un catálogo bimensual que incluye más de 200 programas para Commodore 64 y Spectrum.

— Podrás adquirir programas con un 30% de descuento, por ser miembro del CLUB DE SOFT DE ESPAÑA.

— Obtendrás descuentos de hasta un 20% sobre hardware.



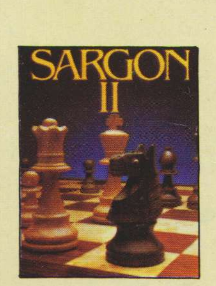
¿COMO PUEDES HACERTE SOCIO DEL CLUB DE SOFT DE ESPAÑA?

— Sólo tendrás que abonar 1.000 Ptas. como cuota UNICA de inscripción.

— Como regalo de bienvenida a nuestro CLUB de SOFT de ESPAÑA podrás elegir entre dos programas:

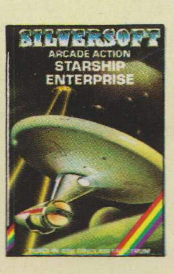
AJEDREZ	STARSHIP ENTERPRISE
SOLO FLIGHT	SLIPPERY SID
para CBM-64	para SPRECTRUM

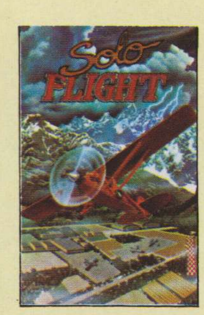
Te enviaremos el programa elegido completamente GRATIS, junto con la confirmación de tu pertenencia al CLUB de SOFT de ESPAÑA.





CLUB de SOFT de ESPAÑA





DIFUSION POR AMISTAD

Habla a tus amigos de las enormes ventajas que supone pertenecer al CLUB DE SOFT DE ESPAÑA. Si junto con tu inscripción se inscriben dos amigos tuyos por tu recomendación, te **regalamos** un programa de juegos.



CUPON DE INSCRIPCION AL CLUB DE SOFT DE ESPAÑA

Eraso, 12 - Tel. 24610 94 - 28028-MADRID

Nombre ____________________

Dirección ____________________

Ciudad ____________________ Tel. ____________________

Deseo pertenecer al CLUB DE SOFT DE ESPAÑA. Ruego me envíen como regalo de bienvenida al CLUB el programa ____________________

Firma

El CLUB DE SOFT DE ESPAÑA te enviará el programa solicitado, junto con un reembolso de 1.000 Ptas. más gastos de envío, como cuota de inscripción al Club.

DIFUSION POR AMISTAD

NOMBRE de un amigo ____________________

Dirección ____________________

Programa de regalo de bienvenida ____________________

Firma

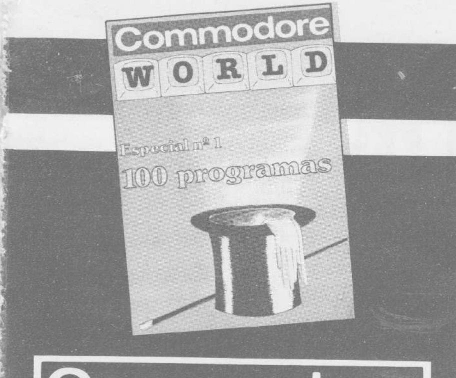
DIFUSION POR AMISTAD

NOMBRE de un amigo ____________________

Dirección ____________________

Programa de regalo de bienvenida ____________________

Firma





Commodore World
está publicado por SIMSA
y la colaboración
de todos nuestros lectores.

EQUIPO
Manuel AMADO; Nieves CHESA;
José Luis ERRAZQUIN;
M.ª Jesús GARCIA;
Miguel A. HERMOSELL;
Alvaro IBAÑEZ; María LOPEZ;
Juan MARQUEZ;
Juan MARTINEZ; Pere MASATS;
Victoria MORALES; Rafael PARDO;
Diego ROMERO; Albert SANGLAS;
Jordi SASTRE; Valerie SHANKS;
Francisco ZABALA...
...Y NUESTROS LECTORES

SIMSA
Coordinador María López
c/Barquillo, 21-3º. Izda. 28004
Madrid. Teléf.: (91) 231 23 88/95
Delegación en Barcelona:
Sant Gervasi de Cassoles, 39
despacho 4. 08022 Barcelona
Teléfs.: (93) 212 73 45/212 88 48

Colabora MEC-COMMODORE
con Club Commodore
Coordinador Pere Masats
c/Valencia, 49-53
08015 Barcelona
Teléf.: (93) 325 50 08

Imprime:
IBERDOS, S. A.
Germán Pérez Carrasco, 24.
28027 Madrid
Depósito Legal: M-2944-1984

PRESENTACION

OS damos la bienvenida a nuestro primer número especial que encima coincide con las fiestas navideñas.
Este número extra, sí que se puede decir que es completamente vuestro, pues sois vosotros y vuestra gran cantidad de colaboraciones lo que dieron nacimiento a la idea.

Habiendo prometido desde el principio que se publicarían todas las colaboraciones que se recibieran y fueran válidas técnicamente y, habiéndonos encontrado con más de cincuenta a las que era físicamente imposible dar salida en los números mensuales de la revista, teníamos que inventarnos algo, máxime cuando más colaboraciones seguían y siguen llegando a diario.

¡Y... así nació el extraordinario! Y... una vez nacida la idea, ¿por qué no acompañaros durante estas vacaciones para que, aparte de nuestra cita mensual que atrae cada vez a más "Commodorianos", sigáis pegando a las teclas y no se os atrofien los dedos?

Sí, pero, y del concurso y premios... ¿qué?

VALE, desde el principio vimos claro que era sólo, de la más elemental justicia, que todos los que hayan hecho posible este número recibieran una "participación" sobre el mismo, por lo que hemos efectuado un sorteo especial para los participantes de este número y a todos les ha correspondido una participación económica de diversas cantidades y algún otro premio adicional.

Pero, ¿quiénes iban a entrar en el mismo y quiénes en la revista mensual? Bueno, eso era fácil —para algo tiene que servir la informática—. Todos al ordenador y, aleatoriamente, quien cayera. Y así funcionó la cosa. La suerte decidió el tema.

Nos gustaría sinceramente poder sacar una publicación extraordinaria de vez en cuando y, aunque tenemos otras cosas e ideas en cartera, os invitamos a seguir mandando vuestras colaboraciones y agradecemos el esfuerzo que estáis haciendo.

Como no es nuestra intención estorbar (lo de la presentación siempre es un poco pesado) y como sabemos que estaréis deseando empezar con el tecleo, nos despedimos de vosotros con mucho cariño y esperamos que os guste el Primer Número Extraordinario de "Commodore World".



Commodore World es miembro de CW Communications/Inc., el grupo editorial más grande y de mayor prestigio en el mundo en lo que se refiere al ámbito informático. Dicho grupo tiene a su cargo la edición de 52 publicaciones relacionadas con los ordenadores en 19 países. El grupo editorial está integrado por:
ALEMANIA: Computerwoche, Micro Computerwelt, PC Welt, Software Markt, CW Edition/Seminar, Computer Business, Run (Commodore). **ARABIA SAUDI:** Saudi Computerworld. **ARGENTINA:** Computerworld/Argentina. **AUSTRALIA:** Australasian Computerworld, Micro Computer Magazine, PC World. **BRASIL:** DataNews, Micro-Mundo. **DINAMARCA:** Computerworld/Danmark, Micro Verden. **ESPAÑA:** Computerworld/España. MicroSistemas, Commodore World. **ESTADOS UNIDOS:** Computerworld, Hot CoCo, InCider, InfoWorld, Microcomputing, PC World, 80-Micro, Mac World Micro Market Word, PC Jr. World Run (Commodore). **FINLANDIA:** Mikro. **FRANCIA:** Le Monde Informatique. **HOLANDA:** Computerworld Benelux, Micro/Info. **ITALIA:** Computerworld Italia. **JAPON:** Computerworld Japan, Perso Comworld, PC World. **MEXICO:** Computerworld/México, Compumundo. **NORUEGA:** Computerworld Norge, Mikro Data. **REINO UNIDO DE GRAN BRETAÑA:** Computer Management. Computer Business Europe. **REPUBLICA POPULAR DE CHINA:** China Computerworld. **SINGAPUR:** Asian Computerworld. **SUECIA:** ComputerSweden. MikroDatorn, Min Hemadator.

SUMARIO

El reino: juego de aventuras

Run. EE.UU.

VIC-20
C-64



Este juego es divertido y educativo a la vez. Tienes que estar haciendo cálculos de números para poder jugar bien, pero no te preocupes que no te va a resultar doloroso. (El programa funciona en el C-64 y en el VIC-20 ampliado o sin ampliar. Ver listados 1 y 2).

Tú eres el dueño y señor de un pequeño estado de hace muchísimos años. Tu meta principal es aumentar el tamaño de tu reino, y el progreso que haces se mide en la cantidad de acres que posees. Para tener éxito en tu propósito, te darás cuenta que preocuparte de tus súbditos con cariño y amor no siempre resulta productivo. Sin embargo, una actitud de desinterés total hacia la población también trae sus riesgos.

Decisiones reales. Cuando empieza el juego tú tienes unos terrenos, tienes almacenada una cantidad de grano y también tienes que gobernar la población. En cada jugada (que dura un año), tienes que comprar o vender tierras, reservar una cantidad de grano para alimentar a la población y decidir la cantidad que se siembra.

Hay muchos factores que tienes que tener en cuenta. El reino puede estar en estado de paz o de guerra. Tienes que decidir si es mejor alimentar a la población o si algunas personas tendrán que morir de hambre. Podrías necesitar que trabajen un número de personas en el campo y otras que vayan a luchar en la guerra. El precio de las tierras puede haber subido o bajado. Tienes que calcular el grano que tienes disponible para sembrar.

Para que tu trabajo sea más fácil de llevar, recibes constantemente un informe sobre el estado del reino que te indicará todos los datos actualizados para que puedas tomar decisiones cara al año siguiente. Además, al final de cada año, el secretario de estado te mandará un informe sobre los resultados de tus decisiones, incluyendo cosas como la cosecha, un censo sobre la población y la economía.

Podrías descubrir, a base de hacer pruebas, las necesidades que son decisivas a la hora de distribuir el grano, pero esto resulta un poco laborioso. En vez de esto, te diré lo siguiente: cada persona requiere 20 arrobas de grano para alimentarse; cada persona puede sembrar solamente 10 acres de tierra; necesitas 1/2 arrobas para sembrar un acre.

En estado de guerra, una tercera parte de la población tiene que alistarse en el ejército y por lo tanto no está disponible para sembrar (sin embargo, estas personas necesitan comer). El reino estará en estado de guerra aproximadamente el 30 por ciento del tiempo.

Puedes variar tu estilo de gobernar, pero yo encuentro que resulta más fácil conseguir tu meta comprando y vendiendo tierras en el momento más oportuno. A veces tendrás que vender tierras para alimentar a la poblacón, y otras veces será necesario que algunas personas se mueran de hambre. La vida era muy dura en aquel entonces.

C-64

```
10 REM GOBIERNA TU REINO
12 POKE53281,1:PRINT"[BLK]";CHR$(14)

14 P=100:AR=10:SP=0:RD=150:YH=3:GH=3
000:CA=1000:YR=1:GS=2850
16 GOTO114
18 REM SUBRUTINAS
20 PRINT"[CLR][10SPC][RVSON][2SPC][SHIFTE]
STADO[SPC]DEL[SPC]REINO[2SPC][RVSOFF]
":PRINT
22 PRINTSPC(9);"[BLK][RVSON][SHIFTP]
OBLACION[7SPC]:";P
24 PRINTSPC(9);"[BLK][RVSON][SHIFTA]
CRES[11SPC]:";CA
26 PRINTSPC(9);"[RED][RVSON][SHIFTA]
RR.[SPC][SHIFTA]LMACENADAS:";GS
28 PRINTSPC(12);"[BLK][CRSRD][SHIFTE]
L[SPC][SHIFTR]EINO[SPC]ESTA[SPC]EN[SPC]
";PW$
30 PRINT:PRINT"[RVSON][39SPC][RVSOFF]
":PRINT:RETURN
32 I=VAL(I$):I=INT(ABS(I)):RETURN
34 REM LAZO DE RETARDO
36 FORX=1TO4000:NEXT:RETURN
38 REM ESPERA UNA TECLA
40 PRINTSPC(5);"[CRSRD][SHIFTP]ULSA[SPC]
UNA[SPC]TECLA[SPC]PARA[SPC]COMENZAR"

42 GETA$:IFA$=""THEN42
44 RETURN
46 PRINT"[CLR][RVSON]*[SPC][SHIFTS]A
LUDOS[SPC][SHIFTG]RAN[SPC][SHIFTR]EY
[SPC]![SPC]*[RVSOFF][CRSRD]"
48 PRINT"[SHIFTH]E[SPC]DE[SPC]CONTAR
TE[SPC]QUE[SPC]EN[SPC]EL[SPC]AN[CRSRL]
[CRSRU][COMM@][CRSRD]O[RVSON]";YR;"[RVSOFF]
DE"
50 PRINT"TU[SPC]REINADO:[CRSRD]"
52 IFSP=1THEN58
54 IFSP=0THENPRINT"[SPC][SHIFTN]ADIE
[SPC]MURIO[SPC]DE[SPC]HAMBRE,Y[SPC]"
;:GOTO60
56 PRINTSP;"PERSONAS[SPC]MURIERON[SPC]
DE[SPC]HAMBRE,Y[SPC]";
57 GOTO60
58 PRINT"[SHIFTU]NA[SPC]PERSONA[SPC]
MURIO[SPC]DE[SPC]HAMBRE,Y[SPC]"
60 IFWK<1THEN68
62 IFWK=1THENPRINT"[SHIFTU]NA";:GOTO
66
64 PRINTWK;
66 PRINT"[SPC]MURIERON[SPC]EN[SPC]BA
TALLA"
68 IFAR=1THEN72
70 PRINTAR;"[SHIFTP]ERSONAS[SPC]VINI
ERON[SPC]AL[SPC]REINO":GOTO76
72 IFSP<10ANDSP>1ANDWK<1THENPRINT"[SPC]
";
74 PRINT"[SHIFTU]NA[SPC]PERSONA[SPC]
VINO[SPC]AL[SPC]REINO."
76 PRINT"[CRSRD][SHIFTR]ECOLECTAMOS"
;GH
78 PRINT"[SHIFTA]RROBAS[SPC]A[SPC]RA
ZON[SPC]DE";YH;"[SHIFTA]RROBAS"
80 PRINT"POR[SPC]ACRE."
82 PRINT"[SHIFTL]AS[SPC]RATAS[SPC]DE
STRUYERON";RD;"ARROBAS."
```

```
84 PRINT"[SHIFTD]EJANDO";GS
86 PRINT"ARROBAS[SPC]ALMACENADAS."
88 PRINT"[CRSRD][SHIFTE]L[SPC]REINO[SPC]
TIENE";CA
90 PRINT"ACRES[SPC]DE[SPC]TIERRA."
92 RETURN
94 PRINT"[CRSRD][SHIFTO][SHIFTH][SPC]
[SHIFTR]EY[SPC]...PIENSALO[SPC]OTRA[SPC]
VEZ!":GOSUB36:RETURN
96 PRINT"[CRSRD][SHIFTL]OS[SPC]CAMPE
SINOS[SPC]HAMBRIENTOS[SPC]ASALTARON"
:PRINT"LOS[SPC]GRANEROS[SPC]Y[SPC]DE
STRUYERON
98 PRINT"[SPC]EL[SPC]50%[SPC]DEL[SPC]
GRANO[SPC]ALMACENADO"
100 GS=INT(GS/2):WR=0:GOSUB36:RETURN

102 IFJ>0THENRETURN
104 P=P-INT(P/2):PRINT"[SHIFTL]A[SPC]
PLAGA[SPC]MATO[SPC]A[SPC]LA[SPC]MITA
D[SPC]DE[SPC]LA[SPC]POBLACION"
106 PRINT"[SHIFTL]A[SPC]POBLACION[SPC]
ES[SPC]AHORA[SPC]DE";P;".":GOSUB36:R
ETURN
108 IFSP<PTHENRETURN
110 PRINT"[CLR][2CRSRD][SHIFTT]ODOS[SPC]
TUS[SPC]SUDBITOS[SPC]HAN[SPC]MUERTO.
....[6SPC][CRSRD][SHIFTT]U[SPC]TAMBI
EN![2CRSRD]"
112 GOTO236
114 PRINT"[CLR][2CRSRD][9SPC][RVSON]
[3SPC][SHIFTG]OBIERNA[SPC]TU[SPC]REI
NO[2SPC][RVSOFF]"
116 PRINT"[CRSRD][SHIFTE]RES[SPC]EL[SPC]
REY[SPC]DEL[SPC]ANTIGUO[SPC]REINO[SPC]
DE[23SPC][SHIFTB]ABILONIA"
118 PRINT"[CRSRD][SHIFTE]L[SPC]OBJET
IVO[SPC]DEL[SPC]JUEGO[SPC]ES[SPC]LLE
VAR[9SPC]PROSPERAMENTE[SPC]EL[SPC]RE
INO"
120 PRINT"[CRSRD][SHIFTP]ARA[SPC]FIN
ALIZAR,VENDE[SPC]TODA[SPC]TU[SPC]TIE
RRA"
122 GOSUB38
124 ZZ=RND(TI):GOTO128
126 ZZ=RND(1):IFZZ<.3THENPW$="[SHIFTG]
[SHIFTU][SHIFTE][2SHIFTR][SHIFTA]":W
F=1.5:GOTO130
128 WF=1:PW$="[SHIFTP][SHIFTA][SHIFTZ]
"
130 GOSUB46:GOSUB38:GOSUB20
132 K=INT(RND(1)*12+16):PRINT"[SHIFTE]
L[SPC]PRECIO[SPC]DE[SPC]LOS[SPC]TERR
ENOS[SPC]ES[SPC]DE";K
134 PRINT"ARROBAS[SPC]POR[SPC]ACRE.[CRSRD]
"
136 PRINT"[4SPC][RVSON][SHIFTC][RVSOFF]
OMPRA,[RVSON][SHIFTV][RVSOFF]ENDE,O[SPC]
[RVSON][SHIFTM][RVSOFF]ANTEN[SPC]LA[SPC]
TIERRA"
138 GOSUB42
140 IFA$="C"ORA$="V"ORA$="M"THEN144
142 PRINT"[2CRSRL][CRSRU]":GOTO138
144 IFA$="M"THEN172
146 PRINT"[CRSRD][SHIFTC]UANTOS[SPC]
ACRES"
148 INPUT"[2SPC]*[3CRSRL]";I$
150 IFI$="*"THENPRINT"[2CRSRU]":GOTO
148
152 GOSUB32:IFI=0THEN172
154 IFA$="V"THEN164
156 J=I*K:IFJ<=GSTHEN160
158 GOSUB94:GOSUB20:GOTO136
160 GS=GS-J:CA=CA+I
162 GOSUB20:GOTO172
164 IFI=CATHEN234
166 IFI>CATHENGOSUB94:GOSUB20:GOTO14
6
168 CA=CA-I:GS=GS+K*I
170 GOSUB20
172 PRINT:PRINT"[SHIFTC]UANTAS[SPC]A
RROBAS[SPC]PARA[SPC]ALIMENTACION"
174 INPUT"[2SPC]*[3CRSRL]";I$
176 IFI$="*"THENPRINT"[3CRSRU]":GOTO
174
178 GOSUB32
180 IFI<=GSTHEN184
182 GOSUB94:GOSUB20:GOTO172
184 IFI>(5*P)THEN190
186 IFI>GS/4THEN190
188 WR=1:REM ROBO EN LOS GRANEROS
190 GS=GS-I:SP=P-INT(I/20):AR=0:GOSU
B108:IFSP>=0THEN194
192 AR=-SP/2:SP=0
194 IFWR=1THENGOSUB96
196 GOSUB20
198 PRINT"[SHIFTC]UANTOS[SPC]ACRES[SPC]
PARA[SPC]SEMBRAR"
200 INPUT"[2SPC]*[3CRSRL]";I$
202 IFI$="*"THENPRINT"[3CRSRU]":GOTO
200
204 GOSUB32
206 IFI>CATHEN214
208 J=INT(I/2):IFJ>GSTHEN214
210 IFI>(10*P/WF)THEN214
212 GOTO218
214 GOSUB94:GOSUB20:GOTO198
216 REM CALCULA CAMBIOS ANUALES
218 GS=GS-J:YH=1+INT(RND(1)*5):GH=YH
*I:RD=INT((GS+GH)*.07*RND(1)):GS=GS-
RD+GH
220 J=INT(RND(1)*25):IFWF=1.5THENWK=
INT(.3*RND(1)*P):GOTO224
222 WK=0
224 AR=INT(AR+(5-YH)*GS/600+1):IFAR<
=99THEN228
226 AR=99
228 IFAR<0THENAR=0
230 P=P+AR-SP-WK:YR=YR+1
232 GOSUB102:GOTO126
234 PRINT"[CLR][CRSRD][SHIFTH]AS[SPC]
VENDIDO[SPC]TODA[SPC]TU[SPC]TIERRA,E
L[SPC]JUEGO[5SPC]SE[SPC]HA[SPC]ACABA
DO"
236 PRINT"[CRSRD][SHIFTA]BANDONA[SPC]
TU[SPC]REINADO."
238 CA=CA+INT(GS/K):B$="[SHIFTM]EJOR
ASTE"
240 IFCA<1130THENB$="[SHIFTE]MPEORAS
TE"
242 PRINT"[CRSRD][SHIFTD]URANTE[SPC]
TO[SPC]REINADO,[SPC]";B$:PRINT"[SPC]
EL[SPC]ESTADO";
244 PRINT"[SPC]DEL[SPC]REINO[SPC]EN[SPC]
UN":PRINTINT((CA/1130)*1000)/10;"%"
246 PRINT"AL[SPC]CABO[SPC]DE"YR;"AN[CRSRL]
[CRSRU][COMM@][CRSRD]OS."
248 END
```

VIC-20

```
10 REM GOBIERNA TU REINO
12 XM=242:POKE36879,93:POKE646,0:IFP
EEK(4096)THENXM=194
14 POKE36869,XM:P=100:AR=10:SP=0:RD=
150:YH=3:GH=3000:CA=1000:YR=1:GS=285
0
16 GOTO114
18 REM SUBRUTINAS
20 PRINT"[CLR][YEL][RVSON][3SPC][SHIFTE]
STADO[SPC]DEL[SPC]REINO[3SPC][RVSOFF]
";
22 PRINT"[CRSRD][WHT][RVSON][SHIFTP]
OBLACION:";P
24 PRINT"[RVSON][SHIFTA]CRES:";CA
26 PRINT"[RVSON][SHIFTA]RR.[SPC][SHIFTA]
LMACENADAS:";GS
```

```
28 PRINT"[BLK][CRSRD][SHIFTE]STAMOS[SPC]
EN[SPC]";PW$
30 PRINT"[RVSON][22SPC][RVSOFF]":RET
URN
32 I=VAL(I$):I=INT(ABS(I)):RETURN
34 REM LAZO DE RETARDO
36 FORX=1TO4000:NEXT:RETURN
38 REM ESPERA UNA TECLA
40 PRINT"[CRSRD][4SPC][SHIFTP]ULSA[SPC]
UNA[SPC]TECLA"
42 GETA$:IFA$=""THEN42
44 RETURN
46 PRINT"[CLR][RVSON]*[SPC][SHIFTS]A
LUDOS[SPC]GRAN[2SPC][SHIFTR]EY[SPC]!
*[RVSOFF]"
48 PRINT"[CRSRD][SHIFTH]E[SPC]DE[SPC]
DECIRTE[SPC]QUE[SPC]EN[2SPC]EL[2SPC]
AN[CRSRL][CRSRU][COMM@][CRSRD]O[SPC]
[RVSON]";YR;"[RVSOFF]DE[SPC]TU"
50 PRINT"REINADO:";
52 IFSP=1THEN58
54 IFSP=0THENPRINT"[SHIFTN]ADIE[SPC]
MURIO[SPC]DE[SPC]HAMBRE,Y":GOTO60
56 PRINTSP;"PERSONAS[SPC]MURIERON[SPC]
DE[2SPC]HAMBRE[SPC]Y[SPC]";:GOTO60
58 PRINT"[SHIFTU]NA[SPC]PERSONA[SPC]
MURIO[SPC]DE[2SPC]HAMBRE[SPC]Y[SPC]"
;
60 IFWK<1THEN68
62 IFWK=1THENPRINT"[SHIFTU]NA";:GOTO
66
64 PRINTWK;
66 PRINT"[SHIFTM]URIERON[SPC]EN[SPC]
BATALLA"
68 IFAR=1THEN72
70 PRINTAR;"PERSONAS[SPC]VINIERON[SPC]
AL[SPC]REINO":GOTO76
72 IFSP<10ANDSP>1ANDWK<1THENPRINT"[SPC]
";
74 PRINT"[SHIFTU]NA[SPC]PERSONA[SPC]
VINO[SPC]AL[SPC]REINO."
76 PRINT"[CRSRD][SHIFTR]ECOLECTAMOS"
;GH
78 PRINT"ARROBAS,[SPC]A[SPC]RAZON[SPC]
DE";YH;"ARROBAS[SPC]POR[SPC]ACRE"
80 REM
82 PRINT"[SHIFTL]AS[SPC]RATAS[SPC]DE
STRUYERON";RD;"[SHIFTA]RROBAS."
84 PRINT"[SHIFTD]EJANDO";GS;"[SHIFTA]
RROBAS"
86 PRINT"ALMACENADAS."
88 PRINT"[CRSRD][SHIFTE]L[SPC]REINO[SPC]
TIENE";CA
90 PRINT"ACRES[SPC]DE[SPC]TIERRA."
92 RETURN
94 PRINT"[CRSRD][SHIFTO][SHIFTH][SPC]
[SHIFTR]EY...PIENSALO[5SPC]DE[SPC]NU
EVO!":GOSUB36:RETURN
96 PRINT"[CRSRD][SHIFTL]OS[SPC]CAMPE
SINOS[8SPC]HAMBRIENTOS[SPC]ATACARON[2SPC]
LOS[SPC]GRANEROS"
98 PRINT"DESTRUYENDO[SPC]EL[SPC]50%[SPC]
DEL[SPC]GRANO[SPC]ALMACENADO"
100 GS=INT(GS/2):WR=0:GOSUB36:RETURN

102 IFJ>0THENRETURN
104 P=P-INT(P/2):PRINT"[SHIFTL]A[SPC]
PLAGA[SPC]MATO[SPC]A[SPC]LA[SPC]MITA
D[SPC]DE[SPC]LA[SPC]POBLACION."
106 PRINT"[SHIFTL]A[SPC]POBLACION[SPC]
ES[SPC]DE";P;".":GOSUB36:RETURN
108 IFSP<PTHENRETURN
110 PRINT"[CLR][2CRSRD][SHIFTT]ODOS[SPC]
TUS[SPC]SUDBITOS[SPC]HAN[SHIFTM]UERT
O......[10SPC][CRSRD][SHIFTT]U[SPC]T
AMBIEN![2CRSRD]"
112 GOTO236
114 PRINT"[CLR][RVSON][3SPC][SHIFTG]
OBIERNA[SPC]TU[SPC]REINO[2SPC][RVSOFF]
"
116 PRINT"[CRSRD][SHIFTE]RES[SPC]EL[SPC]
GOBERNANTE[SPC]DELANTIGUO[SPC]REINO[SPC]
DE[6SPC][SHIFTB]ABILONA"
118 PRINT"[CRSRD][SHIFTE]L[SPC]OBJET
O[SPC]DEL[SPC]JUEGO[SPC]ESMANTENER[SPC]
PROSPERO[SPC]EL[2SPC]REINO"
120 PRINT"[CRSRD][SHIFTS]I[SPC]QUIER
ES[SPC]ACABAR,[4SPC]VENDE[SPC]TODA[SPC]
LA[SPC]TIERRA."
122 GOSUB38
124 ZZ=RND(TI):GOTO128
126 ZZ=RND(1):IFZZ<.3THENPW$="[SHIFTG]
[SHIFTU][SHIFTE][2SHIFTR][SHIFTA]":W
F=1.5:GOTO130
128 WF=1:PW$="[SHIFTP][SHIFTA][SHIFTZ]
"
130 GOSUB46:GOSUB38:GOSUB20
132 K=INT(RND(1)*12+16):PRINT"[SHIFTL]
A[SPC]TIERRA[SPC]ESTA[SPC]A";K
134 PRINT"[SHIFTA]RROBAS[SPC]EL[SPC]
ACRE.[CRSRD]"
136 PRINT"[RVSON][SHIFTC][RVSOFF]OMP
RA,[RVSON][SHIFTV][RVSOFF]ENDE,O[8SPC]
[RVSON][SHIFTM][RVSOFF]ANTEN[SPC]LA[SPC]
TIERRA"
138 GOSUB42
140 IFA$="C"ORA$="V"ORA$="M"THEN144
142 PRINT"[2CRSRL][CRSRU]":GOTO138
144 IFA$="M"THEN172
146 PRINT"[CRSRD][SHIFTC]UANTOS[SPC]
ACRES"
148 INPUT"[2SPC]*[3CRSRL]";I$
150 IFI$="*"THENPRINT"[2CRSRU]":GOTO
148
152 GOSUB32:IFI=0THEN172
154 IFA$="V"THEN164
156 J=I*K:IFJ<=GSTHEN160
158 GOSUB94:GOSUB20:GOTO136
160 GS=GS-J:CA=CA+I
162 GOSUB20:GOTO172
164 IFI=CATHEN234
166 IFI>CATHENGOSUB94:GOSUB20:GOTO14
6
168 CA=CA-I:GS=GS+K*I
170 GOSUB20
172 PRINT:PRINT"[SHIFTC]UANTOS[SPC]A
CRES[SPC]PARA[SPC]ALIMENTACION"
174 INPUT"[2SPC]*[3CRSRL]";I$
176 IFI$="*"THENPRINT"[3CRSRU]":GOTO
174
178 GOSUB32
180 IFI<=GSTHEN184
182 GOSUB94:GOSUB20:GOTO172
184 IFI>(5*P)THEN190
186 IFI>GS/4THEN190
188 WR=1
190 GS=GS-I:SP=P-INT(I/20):AR=0:GOSU
B108:IFSP>=0THEN194
192 AR=-SP/2:SP=0
194 IFWR=1THENGOSUB96
196 GOSUB20
198 PRINT"[SHIFTC]UANTOS[SPC]ACRES[SPC]
PARA[SPC]CULTIVOS"
200 INPUT"[2SPC]*[3CRSRL]";I$
202 IFI$="*"THENPRINT"[3CRSRU]":GOTO
200
204 GOSUB32
206 IFI>CATHEN214
208 J=INT(I/2):IFJ>GSTHEN214
210 IFI>(10*P/WF)THEN214
212 GOTO218
214 GOSUB94:GOSUB20:GOTO198
216 REM CALCULO ANUAL
218 GS=GS-J:YH=1+INT(RND(1)*5):GH=YH
*I:RD=INT((GS+GH)*.07*RND(1)):GS=GS-
RD+GH
220 J=INT(RND(1)*25):IFWF=1.5THENWK=
INT(.3*RND(1)*P):GOTO224
222 WK=0
```

```
224 AR=INT(AR+(5-YH)*GS/600+1):IFAR<
=99THEN228
226 AR=99
228 IFAR<0THENAR=0
230 P=P+AR-SP-WK:YR=YR+1
232 GOSUB102:GOTO126
234 PRINT"[CLR][CRSRD][SHIFTH]AS[SPC]
VENDIDO[SPC]TODA[SPC]TU[3SPC]TIERRA[SPC]
Y[SPC]EL[SPC]JUEGO[SPC]SE[2SPC]HA[SPC]
ACABADO"
236 PRINT"[CRSRD][SHIFTP]UEDES[SPC]A
BANDONAR[SPC]EL[3SPC]REINO."
238 CA=CA+INT(GS/K):B$="[SHIFTM]EJOR
ASTE"
240 IFCA<1130THENB$="[SHIFTE]MPEORAS
TE"
242 PRINT"[CRSRD][SHIFTD]URANTE[SPC]
TU[SPC]REINADO,":PRINTB$;"[SPC]EL[SPC]
ESTADO"
244 PRINT"DEL[SPC]REINO[SPC]EN[SPC]U
N";INT((CA/1130)*1000)/10;"%[SPC]"
246 PRINT"AL[SPC]CABO[SPC]DE";YR;"AN
[CRSRL][CRSRU][COMM@][CRSRD]OS."
248 END
```



Ayuda para teclear los listados

Cuando estamos copiando un programa, tenemos que mirar a tres sitios a la vez: Al teclado, a la pantalla y al listado. Como sólo tenemos dos ojos, esto hace un poco lento el copiar los programas.
Una manera de solucionarlo es grabar en una cinta el programa hablado, y después irlo tecleando a medida que lo vamos oyendo. Además esto es bastante útil para corregir los listados, ya que puede irse revisando mientras lo vamos oyendo.

J.B. RUN EE.UU.

Magia con el cursor

Para hallar en cualquier momento en qué lugar se encuentra el cursor, podemos usar la siguiente subrutina:

```
20 POKE783,PEEK(783)OR1:SYS65520:R=PEEK(781):C=
   PEEK(782):RETURN
```

R es la fila, y C la columna en la que el cursor se encuentra, siendo la primera posición la 0.

B.G.A. RUN EE.UU.

Funciones LOG y EXP.

Las funciones del basic LOG y EXP trabajan en base e, siendo e la constante 2.71828182845904... Nosotros estamos acostumbrados a la base decimal (10). Sin embargo, se puede definir fácilmente un cambio de base del siguiente modo:
DEF FN L(X)=LOG(X)/LOG(10)
Para hallar el logaritmo decimal de un número, ejecuta la orden FN de la siguiente forma:
Variable=FN L(número)
EXP es la función hermana de LOG, calcula el valor de e elevado a la potencia que se encuentra entre paréntesis. Para hacerlo en base 10:
DEF FN E(X)=EXP(X*LOG(10))
Para comprobar que funciona, basta con hacer: PRINT FN L(1000), y nos dará 3, (como todos los estudiantes de matemáticas saben) y PRIN FN E(3) nos dará 1000, pues es la función inversa de la anterior.

P.S. RUN EE.UU.

Ahorrando tiempo

Una manera fácil de ahorrar tiempo cuando queremos hacer un POKE con un valor cero es teclear: POKE XXXX,. El punto es interpretado por el ordenador como un cero, y de este modo ahorramos unos microsegundos en nuestros programas.

R.A.A. RUN EE.UU.

Errores en el cassette

Si encuentras muchos errores en la carga, verificación o grabación de programas, esto puede ser debido (si usas la unidad de discos) a que tengas en memoria el programa WEDGE.
Para solucionarlo, teclea @Q. Esto desconecta el programa y puede que solucione algunos problemas. Para volver a conectarlo teclea SYS52224 y a continuación @8.

A.S. RUN EE.UU.

Carga rápida desde el disco

Cargar un programa desde la unidad de discos 1541 puede ser muy fácil si usas el siguiente truco:
Listas el directorio, mueves el cursor hasta la fila en la que está el programa que quieres cargar; tecleas L[SHIFT 0], (el comando LOAD abreviado); mueves el cursor hasta el final de las comillas y entonces borras la palabra PRG con la barra de espacios. Ahora escribes ,8 (o ,8,1 si es un programa en lenguaje máquina).
Si quieres ahorrarte el borrar PRG, escribe después de las comillas ,8:. Esto hará que el ordenador se olvide de lo que viene a continuación.

C.J. RUN EE.UU.

Abreviaturas

Como seguramente sabrás, el basic commodore ofrece la posibilidad de abreviar algunos de los comandos basic.
Las abreviaturas NO ahorran memoria a la hora de ejecutar un programa, sino que hacen ganar tiempo cuando lo estás tecleando.
Por ejemplo, la orden PRINT puede abreviarse como "?". Ambas formas son codificadas por el ordenador como 153, que es el código de dicha instrucción. Aunque PRINT ocupe 5 caracteres y "?" solo 1, ambos ocupan un solo BYTE de memoria.
Hablando de abreviaturas. Recuerda que no puedes utilizar el símbolo "?" para abreviar el comando PRINT#. La abreviatura correcta de PRINT es P[SHIFT R], que equivale a 152.
Si intentas abreviar PRINT# como "?#", el intérprete basic lo entenderá como un 153 (PRINT) seguido de 35 (el código de "#"), y al ejecutarlo te dará una SYNTAX ERROR.
Lo mismo sucede con las instrucciones GET# e INPUT#, aunque GET# no tiene abreviatura en basic.
I.A. RUN EE.UU.

Instrucciones Basic

Algunas instrucciones basic, realizan por sí mismas la función INT antes de procesar el dato que se les entrega. Este es el caso de las funciones: CHR$, MID$, RIGHT$, LEFT$, PEEK, POKE, SPC, TAB, DIM, GOTO y GOSUB.
Esto quiere decir que no hace falta teclear:
POKE 7680,INT (RND(1)*255)
Basta con poner:
POKE 7680,RND(1)*255

I.A. RUN EE.UU.

Sustituyendo al joystick

Si tienes un C-64 pero no tienes joystick, puedes simularlo desde el teclado. No es lo más conveniente del mundo pero funciona.

	PORT 1	PORT 2
BOTON	espacio	espacio y "M"
ARRIBA	tecla "1"	espacio y "F1"
ABAJO	flecha izq.	espacio y "Z"
IZQUIERDA	tecla CTRL	espacio y "C"
DERECHA	tecla "2"	espacio y "B"

F.E. RUN EE.UU.



Un artista con joystick

Por Michael Platt
(RUN EE.UU.)

C-64
Joystick

¡Deja que tu talento creativo fluya libremente! Con este programa podrás utilizar el joystick y las posibilidades gráficas del C-64 para dibujar, pintar o garabatear a tu antojo.
Las posibilidades que ofrece el Commodore 64 para realizar gráficos en color son verdaderamente sorprendentes. En el modo multicolor, se pueden utilizar hasta cuatro colores distintos en una matriz de 8 × 8 puntos. Lamentablemente, la falta de comandos incorporados en el Basic obliga a utilizar numerosos e incómodos Peeks y Pokes.
Este programa aporta una solución al problema. Ahora podrás dibujar con varios colores desplazando el cursor por la pantalla con un **joystick**. Además, podrás guardar tus obras de arte en cassette o disco.

Variedad de colores. Al ejecutar el programa, el borde será de color azul, y el fondo negro. Estos colores se pueden cambiar con las teclas F5 y F7.
Debes enchufar el joystick en el conector 2. El cursor, que puede desplazarse en cualquier dirección usando el joystick, estará situado en el centro de la pantalla. Para dibujar pulsa el botón de disparo.
El estado del programa se indica en la esquina superior izquierda. En el recuadro izquierdo podrá ver un 1, 2 ó 3 que indican el modo de dibujo en que se encuentra, o una E para indicar el modo de borrar (Erase). El recuadro de la derecha muestra el color en que va a dibujar el cursor. Entre los dos recuadros verás uno o dos puntos que indican si el cursor es normal o doble.
En una matriz de puntos de 8 × 8, sólo se puede utilizar un color para cada modo de dibujo. Para poder utilizar cuatro colores distintos, incluyendo el color de fondo, es necesario emplear los tres modos de dibujo. Con la letra F3 se eligen los modos dibujar o borrar.
El cursor puede dibujar en 16 colores: negro, blanco, rojo, cian, púrpura, verde, azul, amarillo, naranja, marrón, rosa, gris oscuro, gris, gris claro, verde claro y azul claro. Los colores se eligen con la tecla F1.

Las claves del color. El trazo del cursor puede ser fino o grueso. En el modo fino, el cursor utiliza un área de 1 × 2. La tecla de asterisco (*) pasa el cursor de un tipo de trazo al otro. Las teclas numéricas, del uno al nueve, controlan la velocidad del cursor. La tecla número uno es la de mayor velocidad, y la nueve la menor (para trabajos de detalle).

La tecla CLR/home con shift borra toda la pantalla.
La tecla de libra (£) con shift te permite guardar en la memoria los colores de borde y de fondo que más te gusten. Para llamarlos basta con volver a pulsar esta tecla.
Para ver tu dibujo limpio de polvo y paja, pulsa la tecla con el signo menos. Desaparecerá el cursor y el recuadro. Para recuperarlos, pulsa la tecla con el signo más.
Cuando hayas terminado de dibujar, podrás salvar tus diseños en un disco pulsando la tecla F2. Con la tecla F4 los podrás cargar posteriormente. En caso de que utilices un cassette, sálvalos con la tecla F6 y cárgalos con la tecla F8.
Sea cual sea el soporte utilizado, el programa te pedirá que especifiques un nombre para el fichero. Una vez que lo hayas introducido pulsa la tecla Return. Para salir del modo SAVE (Salvar) o LOAD (Cargar), pulsa la tecla Return y luego la tecla run/stop, con lo cual pasará al modo normal Draw.
Cuando salvas un fichero que haya de sustituir a otro fichero del mismo nombre ya existente en el disco, tecla "@0:" antes del nombre del fichero.
En la tabla 1 figura un resumen de los comandos.

Tabla 1. Resumen de las teclas de comando

Tecla	Comando
F1	Cambia color de dibujo del cursor
F3	Cambia modo de dibujo (1, 2, 3 ó E)
F5	Cambia color del borde
F7	Cambia color del fondo
F2	Salva dibujo en disco
F4	Carga dibujo desde el disco
F6	Salva dibujo en cassette
F8	Carga dibujo desde cassette
uno a nueve	Elige velocidad del cursor
Menos	Elimina el cursor y el recuadro
Más	Activa el cursor y el recuadro
CLR Home	Borra la pantalla
Libra mayúscula	Salva los colores de fondo y de borde
Libra	Recupera colores de fondo y de borde
Asterisco	Fija el trazo (grueso o fino) del cursor

El Cargador. Cuando hayas tecleado el programa, primero sálvalo en disco o en cassette, ya que se borrará a sí mismo tras su ejecución. Comprueba que la copia es buena. Teclea RUN para ejecutar el programa, y pulsa la tecla Return. De este modo el programa en lenguaje máquina quedará escrito en las direcciones altas de la RAM $C000-CFFF. Una vez escrito el programa en la memoria, no es necesario volver a cargarlo.
Cuando quieras salir del programa, pulsa la tecla Run/stop y la tecla Restore. Mientras no escribas nada en estas direcciones altas de la RAM, podrá ejecutar un programa Basic sin destruir el código en el lenguaje máquina. No obstante, el programa de gráficos podría afectar a un programa en Basic, por lo que, antes de nada, debes salvarlo en la memoria.
Teclea SYS52992 para volver a activar el programa gráfico, y pulsa la tecla Return.
Antes de finalizar, salva el dibujo que haya en la memoria. Si no lo haces, dicho dibujo se borrará.

La comprobación. Las líneas 20 a 50 contienen una comprobación que te permitirá saber si has tecleado una sentencia data incorrectamente. Una vez introducido y comprobado todo el programa, podrás borrar estas líneas.

Mapas multicolores de bits. Los mapas multicolores de bits permiten crear figuras multicolores muy detalladas. Con este método, se puede visualizar en la pantalla una sección de 8K de memoria. En este caso la sección de memoria ocupa las direcciones $2000-3F3F. Cada dos bits (par de bits) de esta sección de 8K controlan un punto, y hay 32.000 (160×200) puntos en la pantalla.
Las distintas combinaciones de un par de bits determinan el color de cada punto. Con dos bits se pueden hacer cuatro combinaciones de color: con los dos bits a cero (00), con los

dos bits a uno (11), o uno a cero y el otro a uno (01 y 10). En la tabla 2 se detalla de dónde procede la información de color.

Tabla 2. Procedencia de la información de color

Par de bits	La información de color proviene de
00	Color de fondo $D021
01	4 bits más significativos de memoria de pantalla. $0400-07FF
10	4 bits menos significativos de memoria de pantalla $0400-07F
11	4 bits de color $D800-DBFF

Subrutina de puntos. El núcleo de este programa gráfico está constituido por una corta subrutina almacenada en las direcciones $C000-C130, que se podría transportar a otros programas de gráficos de alta resolución.

Antes de utilizarlo, es necesario llamar a la rutina de inicialización que empieza en $C0E7. Dicha rutina activará un punto o un par de puntos en las coordenadas X e Y de la pantalla.

El rango de X va de 0 a 319 (horizontalmente) y el rango de Y va de 0 a 199 (verticalmente). La posición 0,0 se encuentra en la esquina izquierda superior de la pantalla. Antes de entrar en la rutina, la coordenada X debe estar guardada en las posiciones $0340-0341, y la coordenada Y en la posición $0342. Se necesitan dos bytes para la coordenada X porque el rango de X es superior a 255, que es lo máximo que puede contener un solo byte.

Tabla 3. Posiciones de memoria utilizadas

$00FD-00FE	Byte en mapa de bits en que se fija el bit
$0022-0023	Zona de almacenamiento provisional
$0024	Velocidad del cursor
$0025	Cassette o disco
$0026	Trazo doble o sencillo
$0027	Zona de almacenamiento provisional
$003F-0041	Zona de almacenamiento provisional
$00A5	Modo de dibujo
$00A6-00A7	Byte de color
$00A8	Color del cursor
$033C-033F	Zona de almacenamiento provisional
$0340-0341	Coordenada X
$0342	Coordenada Y
$CFFD-CFFF	Zona de almacenamiento provisional
$2000-3F3F	Zona del mapa de bits
$3F45-3F46	Zona reservada para colores de fondo y de borde
$4000-43FF	Zona reservada para mapa de color
$4400-47FF	Zona reservada para mapa de pantalla
$C000-C0E6	Rutina principal
$C0E7-C130	Inicializar mapa de bits
$C150-C1D3	Mover cursor
$C300-C34C	Dibujar
$C400-C4FC	Leer teclas-cambia color de cursor/cambia modo de dibujo/cambia colores de borde y fondo/borra pantalla/cambia velocidad de cursor
$C500-C584	Escribe byte en mapa de color
$C600-C676	Visualización del modo de dibujo
$C700-C742	Leer teclas-cambio de trazo doble o sencillo del cursor
$C900-C96F	Carga
$CA00-CA7A	Almacenamiento
$CB00-CB6E	Inicialización
$CC00-CC4F	Datos para la pregunta de guardar/cargar
$CC50-CCCF	Datos del recuadro de status
$CD00-CD8B	Aceptar nombre para guardar/cargar
$CE00-CE37	Transferir memoria a zona de almacenamiento
$CE50-CE87	Transferir memoria desde zona de almacenamiento
$CF00-CF17	Subrutinas de bucles
$CFA0-CFAB	Transferir datos del recuadro de status a memoria intermedia

Cómo funciona el programa. Para activar un punto hay que encontrar el byte correspondiente en la sección de memoria de 8K. Luego deberá determinar cuál de los 8 bits del byte hay que modificar.

Las fórmulas que determinan los bits y bytes están contenidas en la Guía de Referencia del Programador para el C-64:

Byte=Base+(INT(Y/8)*320)+(INT(X/8)*8)+(Y AND 7)

Bit = 7-(X AND 7)

"Base", en este caso, es igual a 8192, porque allí es donde empieza nuestro mapa de bits. He simplificado la fórmula del byte convirtiéndola en:

Byte=Base+((Y AND 248)*40)+(X AND 248)+(Y AND 7)

Luego, traduje esta fórmula a lenguaje máquina, codificándola en las posiciones $C065-C0C0. Pero, en vez de activar el bit correcto de este byte por la fórmula anterior, utilicé un índice a una tabla de valores de máscara predeterminados. (Una máscara es un conjunto de bits que "extrae" uno o más bits de un grupo de bits). En caso de que quieras variar los valores de máscara y activar bits determinados o pares de bits en la pantalla multicolor, tendrás que desplazar este índice al lugar correcto de la tabla de máscaras.

Las rutinas de almacenamiento y carga de Kernal. Los datos de los dibujos se guardan en un bloque continuo de memoria en $2000-47FF. Antes de salvarlos, los datos de color para el par de bits 11 (por lo general, memoria de color) se transfieren a las direcciones $4000-43FF. Los datos de color para los pares de bits 01 y 10 (generalmente memoria de pantalla) se transfieren a las direcciones $4400-47FF. Al cargar un dibujo los datos se transfieren nuevamente a sus posiciones correctas. Para el almacenamiento y carga de los dibujos, utilicé las subrutinas SETLFS y SETNAM de Kernal.

```
15 PRINT:PRINT"[CLR]CARGANDO[SPC]LEN
GUAJE[SPC]MAQUINA[SPC]:[SPC]ESPERE"
19 REM SUMA DE CONTROL
20 Z=0:S=206122:FORT=0TO1809:READA:Z
=Z+A:NEXT
30 IFZ=STHENPRINT"SENTENCIAS[SPC]DAT
A[SPC]CORRECTAS":GOTO50
40 IFZ<>STHENPRINT"ERROR[SPC]EN[SPC]
LAS[SPC]SENTENCIAS[SPC]DATA":END
50 RESTORE
99 REM RUTINA PRINCIPAL
100 FORT=49152TO49456:READD:POKET,D:
NEXT
110 DATA128,64,32,16,8,4,2,1,192,192
,48,48,12,12,3
120 DATA3,128,128,32,32,8,8,2,2,64,6
4,16,16,4,4,1
130 DATA1,63,63,207,207,243,243,252,
252,169,8,133,34,56,176,29
140 DATA169,16,133,34,56,176,22,169,
24,133,34,56,176,15,169,32
150 DATA133,34,56,176,8,169,0,133,34
,56,176,1,96,173,65,3
160 DATA201,2,176,248,201,1,208,7,17
3,64,3,201,64,176,237,173
170 DATA66,3,201,200,176,230,169,0,1
41,63,3,133,254,173,64,3
180 DATA41,248,141,60,3,173,66,3,41,
7,24,109,60,3,141,60
190 DATA3,173,65,3,105,0,141,61,3,17
3,66,3,41,248,133,253
200 DATA160,5,10,46,63,3,136,208,249
,141,62,3,165,253,160,3
210 DATA10,38,254,136,208,250,24,109
,62,3,133,253,165,254,109,63
```

```
220 DATA3,133,254,24,165,253,109,60,
3,133,253,165,254,109,61,3
230 DATA24,105,32,56,133,254,173,64,
3,41,7,24,101,34,170,189
240 DATA0,192,133,35,160,0,177,253,1
66,34,224,32,240,5,5,35
250 DATA145,253,96,37,35,145,253,96,
162,0,169,38,157,0,4,157
260 DATA0,5,157,0,6,157,0,7,169,1,15
7,0,216,157,0,217
270 DATA157,0,218,157,233,218,232,20
8,225,169,63,133,167,169,0,133
280 DATA166,168,145,166,136,192,0,20
8,249,198,167,166,167,224,31,208
290 DATA241,173,17,208,9,32,141,17,2
08,173,24,208,9,8,141,24
300 DATA208,96
309 REM RUTINA DE MOVIMIENTO DEL CUR
SOR
310 FORT=49488TO49619:READD:POKET,D:
NEXT
320 DATA230,63,165,63,197,36,208,123
,169,0,133,63,173,0,220
330 DATA41,1,208,10,173,1,208,201,40
,240,3,206,1,208,173,0
340 DATA220,41,2,208,10,173,1,208,20
1,239,240,3,238,1,208,173
350 DATA0,220,41,8,208,32,173,16,208
,201,1,208,7,173,0,208
360 DATA201,62,240,18,238,0,208,238,
0,208,173,0,208,201,0,208
370 DATA5,169,1,141,16,208,173,0,220
,41,4,208,20,173,16,208
380 DATA201,0,208,7,173,0,208,201,0,
240,6,206,0,208,206,0
390 DATA208,173,16,208,201,1,208,12,
173,0,208,201,254,208,5,169
400 DATA0,141,16,208,96
409 REM RUTINA DE DIBUJO
410 FORT=49920TO49996:READD:POKET,D:
NEXT
420 DATA173,16,208,141,65,3,173,0,20
8,141,64,3,173,1,208
430 DATA56,233,40,141,66,3,173,0,220
,41,16,208,48,165,165,201
440 DATA0,208,9,32,61,192,206,66,3,3
2,61,192,165,165,201,1
450 DATA208,3,32,40,192,165,165,201,
2,208,6,32,61,192,32,47
460 DATA192,165,165,201,3,208,6,32,6
1,192,32,54,192,96
469 REM LEE EL TECLADO / RUTINAS PAR
A HACER POKES DE COLOR
470 FORT=50176TO50564:READD:POKET,D
NEXT
480 DATA169,8,133,37,32,159,255,32,2
28,255,201,133,208,12,230
490 DATA168,166,168,224,16,208,4,162
,0,134,168,166,168,142,37,208
500 DATA201,134,208,12,198,165,166,1
65,224,255,208,4,162,3,134,165
510 DATA201,135,208,15,238,32,208,17
4,32,208,224,16,208,5,162,0
520 DATA142,32,208,201,136,208,15,23
8,33,208,174,33,208,224,16,208
530 DATA5,162,0,142,33,208,201,49,20
8,4,162,5,134,36,201,50
540 DATA208,4,162,10,134,36,201,51,2
08,4,162,16,134,36,201,52
550 DATA208,4,162,21,134,36,201,53,2
08,4,162,32,134,36,201,54
560 DATA208,4,162,48,134,36,201,55,2
08,4,162,64,134,36,201,56
570 DATA208,4,162,112,134,36,201,57,
208,4,162,255,134,36,201,147
580 DATA208,3,32,0,193,201,137,208,3
,32,0,202,201,138,208,3
590 DATA32,0,201,201,45,208,5,162,0,
142,21,208,201,43,208,5
600 DATA162,3,142,21,208,201,139,208
,7,162,1,134,37,32,0,202
610 DATA201,140,208,7,162,1,134,37,3
2,0,201,201,169,208,12,174
620 DATA33,208,142,254,207,174,32,20
8,142,255,207,201,92,208,12,174
630 DATA254,207,142,33,208,174,255,2
07,142,32,208,133,39,96,2,0
640 DATA0,173,0,220,41,16,208,125,32
,21,195,165,254,41,7,170
650 DATA165,254,74,74,74,133,167,165
,253,74,74,74,133,166,224,0
660 DATA240,15,202,24,165,166,105,32
,133,166,144,242,230,167,56,176
670 DATA237,160,0,165,165,201,3,208,
16,177,166,41,15,106,106,106
680 DATA106,5,168,42,42,42,145,16
6,165,165,201,2,208,8,177
690 DATA166,41,240,5,168,145,166,165
,165,201,1,208,40,24,165,167
700 DATA105,212,133,167,165,253,41,1
176,11,177,166,41,240,5,168
710 DATA145,166,56,176,16,177,166,41
,15,106,106,106,106,5,168,42
720 DATA42,42,42,145,166,96
729 REM RUTINA DE DIBUJO EN LA PANTA
LLA
730 FORT=50688TO50806:READD:POKET,D:
NEXT
740 DATA165,165,201,3,208,25,169,171
,141,201,3,141,204,3,141
750 DATA210,3,141,213,3,169,191,141,
216,3,141,198,3,141,207,3
760 DATA201,2,208,27,169,191,141,198
,3,141,207,3,141,216,3,169
770 DATA171,141,201,3,141,204,3,169,
186,141,210,3,141,213,3,201
780 DATA1,208,23,169,171,141,198,3,1
41,201,3,141,204,3,141,207
790 DATA3,141,210,3,141,213,3,141,21
6,3,201,0,208,25,169,191
800 DATA141,198,3,141,207,3,141,216,
3,169,186,141,201,3,141,210
810 DATA3,141,204,3,141,213,3,96
819 REM RUTINA DEL CURSOR SIMPLE O D
OBLE
820 FORT=50944TO51010:READD:POKET,D:
NEXT
830 DATA165,39,201,42,208,15,166,38,
240,7,162,0,134,38,56
840 DATA176,4,162,1,134,38,166,38,22
4,1,208,25,206,66,3,32
850 DATA0,197,162,140,142,211,3,142,
214,3,162,136,142,217,3,142
860 DATA220,3,56,176,14,162,128,142,
211,3,142,214,3,142,217,3
870 DATA142,220,3,96
879 REM RUTINA DE CARGA
880 FORT=51456TO51567:READD:POKET,D:
NEXT
890 DATA32,0,206,165,165,141,253,207
,169,27,141,17,208,169,21
900 DATA141,24,208,173,33,208,141,69
,63,173,32,208,141,70,63,169
910 DATA0,141,21,208,169,6,141,33,20
8,160,0,185,0,204,32,210
920 DATA255,200,192,40,208,245,165,3
7,201,1,208,13,160,0,185,40
930 DATA204,32,210,255,200,192,20,20
8,245,32,0,205,169,0,32,213
940 DATA255,173,69,63,141,33,208,173
,70,63,141,32,208,169,3,141
950 DATA21,208,32,32,193,32,80,206,3
2,160,207,173,253,207,133,165
960 DATA96
969 REM RUTINA DE SALVAR EN CINTA/DI
SCO
970 FORT=51712TO51834:READD:POKET,D:
NEXT
```

```
980 DATA32,0,206,165,165,141,253,207
,169,27,141,17,208,169,21
990 DATA141,24,208,173,33,208,141,69
,63,173,32,208,141,70,63,169
1000 DATA0,141,21,208,169,6,141,33,2
08,160,0,185,0,204,32,210
1010 DATA255,200,192,20,208,245,165,
37,201,1,208,13,160,0,185,40
1020 DATA204,32,210,255,200,192,20,2
08,245,32,0,205,169,17,32,210
1030 DATA255,169,0,133,61,169,32,133
,62,162,0,160,72,169,61,32
1040 DATA216,255,173,69,63,141,33,20
8,169,3,141,21,208,32,32,193
1050 DATA32,80,206,32,160,207,173,25
3,207,133,165,96
1059 REM INICIALIZACION
1060 FORT=51968TO52078:READD:POKET,D
:NEXT
1070 DATA32,231,192,169,3,141,21,208
,169,1,141,39,208,169,160
1080 DATA141,0,208,141,1,208,169,0,1
41,33,208,169,1,141,29,208
1090 DATA173,22,208,9,16,141,22,208,
169,3,133,165,169,3,141,28
1100 DATA208,169,32,133,36,141,2,208
,169,58,141,3,208,169,6,133
1110 DATA167,169,108,133,166,169,38,
133,170,169,1,133,171,169,1,133
1120 DATA168,169,1,141,40,208,169,0,
133,38,141,37,208,169,6,141
1130 DATA32,208,32,160,207,169,14,14
1,248,7,169,15,141,249,7,96
1149 REM DATAS PARA MENSAJES DE LOAD
/SAVE
1150 FORT=52224TO52431:POKET,0:NEXT
1160 FORT=52224TO52283:READA:POKET,A
:NEXT
1170 DATA147,5,17,17,18,29,29,68,73,
83,75,32,83,65,86
1180 DATA69,32,32,146,58,32,13,145,2
9,29,29,29,29,29,29,18
1190 DATA76,79,65,68,29,29,29,29,146
,19,17,17,18,29,29,84
1200 DATA65,80,69,29,29,29,29,29,29,
29,29,29,146
1210 POKE52329,12:POKE52332,8:POKE52
338,8:POKE52341,12
1219 REM DATAS DE LOS SPRITES
1220 FORT=52371TO52397:READA:POKET,A
:NEXT
1230 DATA170,140,85,191,140,85,171,1
36,85,171,136,85
1240 DATA191,128,85,171,128,85,171,1
28,85,191,128,85,170,128,85
1249 REM SUBRUTINA DE ENTRADA DEL NO
MBRE EN LOAD/SAVE
1250 FORT=52480TO52619:READA:POKET,A
:NEXT
1260 DATA162,0,160,0,232,224,255,208
,251,200,192,16,208,246,32
1270 DATA159,255,32,228,255,201,32,1
44,7,201,96,176,3,32,210,255
1280 DATA201,20,208,3,32,210,255,166
,211,224,31,208,4,160,30,132
1290 DATA211,224,14,208,4,160,15,132
,211,201,13,208,3,56,176,3
1300 DATA56,176,190,169,1,166,37,160
,1,32,186,255,162,112,160,254
1310 DATA202,200,189,0,4,201,32,240,
247,132,65,169,16,56,229,65
1320 DATA168,162,0,24,189,94,4,201,4
7,176,10,201,32,176,6,24
1330 DATA105,64,157,94,4,232,224,17,
208,233,152,162,95,160,4,32
1340 DATA189,255,169,13,32,210,255,1
69,192,32,144,255,96
1349 REM RUTINA PARA TRANSFERIR MEMO
RIA AL AREA DE ALMACENAMIENTO
1350 FORT=52736TO52791:READ:POKET,A:
NEXT
1360 DATA162,0,189,0,216,157,0,64,18
9,0,217,157,0,65,189
1370 DATA0,218,157,0,66,189,0,219,15
7,0,67,189,0,4,157,0
1380 DATA68,189,0,5,157,0,69,189,0,6
,157,0,70,189,0,7
1390 DATA157,0,71,232,224,0,208,203,
96
1399 REM RUTINA PARA TRANSFERIR MEMO
RIA DESDE EL AREA DE ALMACENAMIENTO
1400 FORT=52816TO52871:READA:POKET,A
:NEXT
1410 DATA162,0,189,0,64,157,0,216,18
9,0,65,157,0,217,189
1420 DATA0,66,157,0,218,189,0,67,157
,0,219,189,0,68,157,0
1430 DATA4,189,0,69,157,0,5,189,0,70
,157,0,6,189,0,71
1440 DATA157,0,7,232,224,0,208,203,9
6
1449 REM SUBRUTINAS DE LLAMADA A LOS
 BUCLES
1450 FORT=52992TO53015:READA:POKET,A
:NEXT
1460 DATA32,0,203,32,80,193,32,0,195
,32,0,196,32,0,197
1470 DATA32,0,198,32,0,199,76,3,207
1479 REM RUTINA PARA TRANSFERIR DATA
S AL BUFFER DEL CASSETTE
1480 FORT=53152TO53163:READ:POKET,A:
NEXT
1490 DATA160,128,185,79,204,153,127,
3,136,208,247,96
1500 POKE2049,0:POKE2050,0:SYS52992
```



Truco GET

Algunas veces sucede que se quiere detener un momento la imagen que hay en la pantalla (por ejemplo durante las instrucciones). Pero si uno ya se las sabe es bastante aburrido tener que esperar a que transcurra el FOR-NEXT que seguramente habremos puesto. Otra solución es:

100 for i=1 to 500: get a$: if a$="" then next
110 Aquí sigue el programa.

Estas líneas esperan cierto tiempo a menos que se pulse una tecla.

S.T. RUN EE.UU.

Idea REM

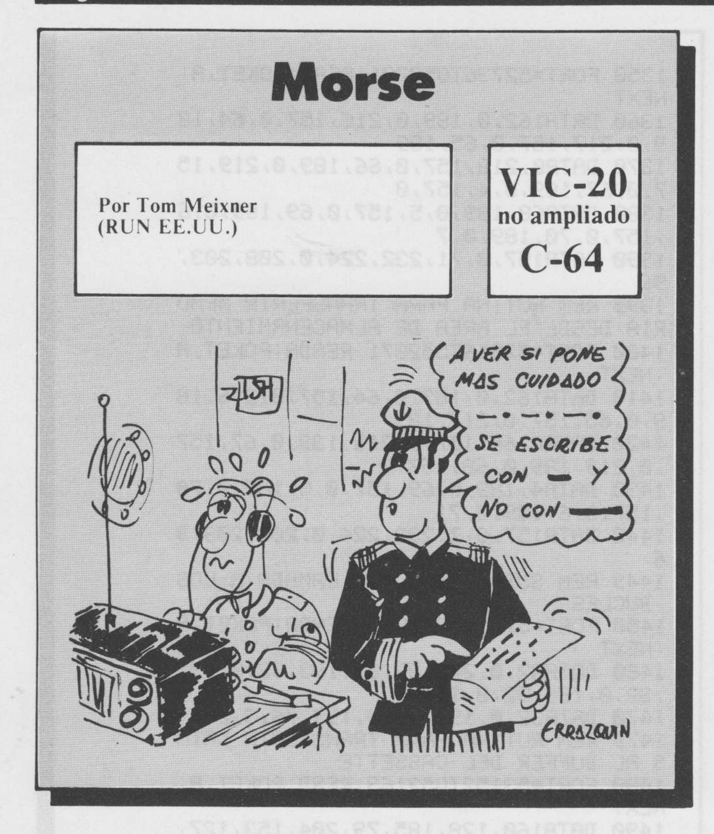
Según va aumentando la biblioteca de programas, va siendo más difícil recordar cuándo se hizo tal o cuál programa. Para solucionarlo nada más fácil que poner una sentencia REM al principio del mismo que diga más o menos:
10 REM —COMMODORE WORLD. NOVIEMBRE 84, PAGINA 76—

C.L.M. RUN EE.UU.

Líneas perdidas

Si accidentalmente borras una línea de un programa, pero la línea en cuestión está todavía en la pantalla, para recuperarla sólo tienes que moverte con el cursor hasta situarte encima de la línea y pulsar RETURN.

G.D. RUN EE.UU.



Programa de código morse para perfeccionar tus puntos y rayas.

Si eres radioaficionado o te gustaría serlo, dominar el código Morse te es indispensable. Este programa, que enseña escuchando, ayudará al principiante a aprender y permitirá al que ya tiene conocimientos, aumentar su velocidad y precisión.

Todos los radioaficionados saben que el primer paso para llegar a ser un verdadero "aficionado" consiste en aprender a enviar y recibir mensajes en Morse rápida y precisamente.

El programa de Código Morse, que se puede ejecutar tanto en el VIC-20 no ampliado como en el Commodore 64, parte de la base de que si tu oyes realmente las secuencias de los sonidos del código Morse, en vez de tratar de memorizar las tablas de puntos y rayas, aprenderás más rápidamente. Partiendo de esta premisa, he desarrollado tres modos distintos en el programa: Práctica, Examen y Texto.

Los Modos. El modo Práctica es muy sencillo. Después de cargar el programa, pulsa P (de Práctica), elige una velocidad y luego pulsa cualquier otra letra o número; el altavoz de la televisión producirá la secuencia de código Morse correspondiente. Simultáneamente, la letra o número que hayas pulsado aparecerá en la esquina izquierda superior de la pantalla. Para salir de este modo, cambiar de velocidad o abandonar el programa, pulse F5, F3 ó F1 respectivamente.

El modo Examen se parece al modo anterior pero a la inversa.

Tú recibes breves instrucciones y eliges la velocidad.

Inmediatamente después, el programa elige al azar una letra o un número y envía el sonido del código Morse al altavoz del televisor. A continuación tú deberás identificar el sonido pulsando una tecla. (Si te equivocas en la pantalla aparecerá el carácter correcto). Mientras tanto, el programa va sumando tus puntos; cada diez ejercicios aparece el resultado en la pantalla y el programa te da la oportunidad de volver a intentarlo.

Luego, pulsando Y, vuelve a aparecer el menú. Al pulsar N se detiene el programa.

El modo Texto es una combinación de los dos modos anteriores. Tras elegir T y una velocidad, aparece una nueva pantalla con instrucciones. En ese momento puedes teclear un texto. Dicho texto, en la versión para el VIC, puede ocupar hasta dos líneas (44 caracteres), mientras que en la versión para el Commodore 64 puede llegar a tres líneas (120 caracteres).

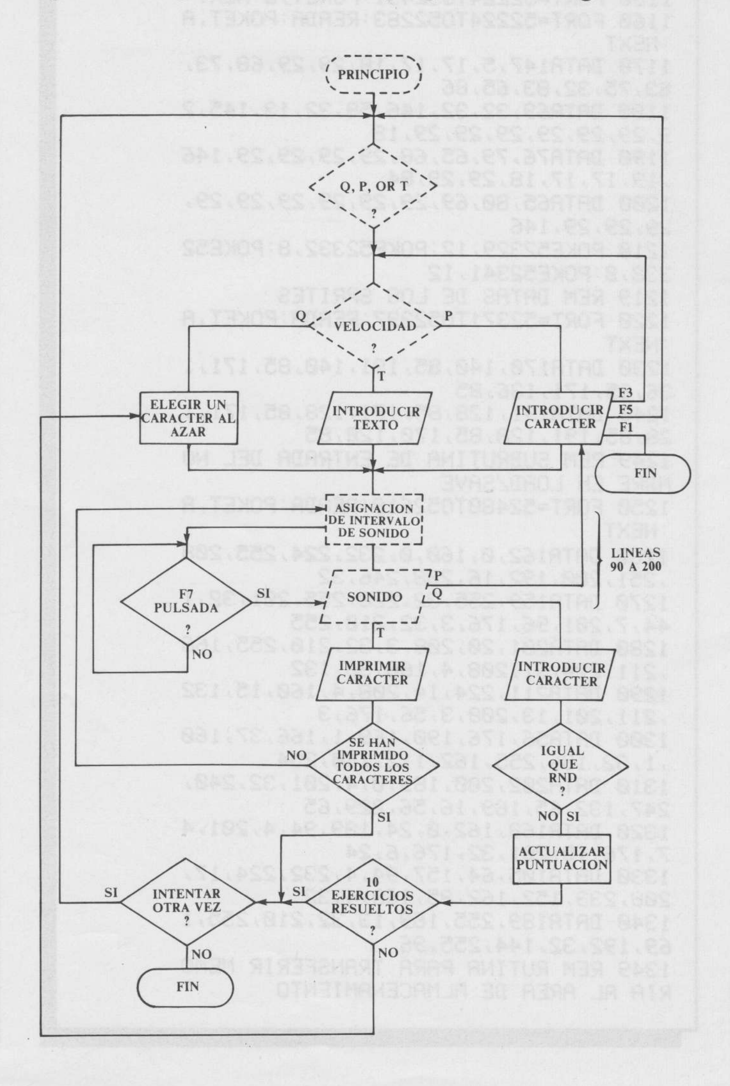
En este momento el texto no se ve en la pantalla, pero el número de caracteres tecleados aparece en una esquina de la misma. Al pulsar la tecla return, o al llegar al límite máximo de texto, se produce la secuencia de texto en código Morse, para que tú la descifres. Cuando quieras comprobar tu traducción pulsa F7, con lo que el texto aparecerá escrito en la pantalla.

Los tres modos combinados constituyen un excelente método para llegar a dominar el código Morse. El modo Práctica permite asociar los caracteres con las secuencias sonoras de puntos y rayas; el modo Examen asocia las secuencias sonoras de puntos y rayas con los caracteres (lo cual es totalmente distinto); y el modo Texto prepara al usuario para emisiones de radio de verdad. Puesto que cada modo ofrece una selección de siete velocidades, el programa puede ser utilizado con provecho tanto por el principiante como por el operador avanzado.

El programa. El secreto de la eficiencia de este programa reside en que los tres modos utilizan las mismas subrutinas: las líneas 5-13, 60-83 y 90-200 (¡la mayor parte del programa!).

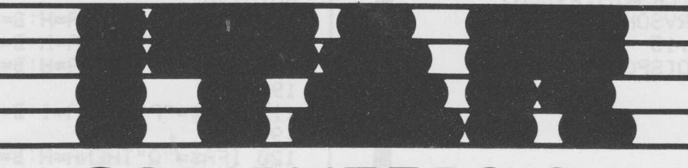
Para comprender mejor cómo funciona el programa, estudia el diagrama de flujo (fig. 1). Los recuadros hechos con líneas de puntos constituyen el núcleo del programa y son, en la práctica, las subrutinas utilizadas en los tres modos. Las líneas y recuadros continuos representan el camino que sigue cada modo durante el curso del programa.

Figura 1: Diagrama de flujo del programa de código Morse



8 MESES DE GARANTIA PARA ORDENADORES Y PERIFERICOS

Y PARA COMERCIOS CONDICIONES INTERESANTISIMAS



COMPUTERS, S.A.

EXPOSICIONES:

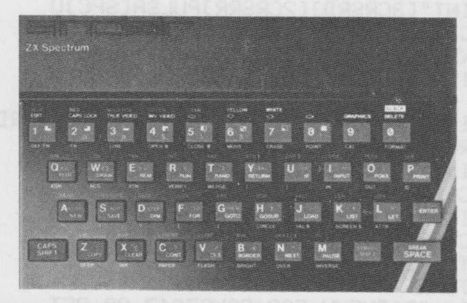
PAMPLONA: C/ Alfonso el Batallador, 16 (trasera) - Tel. 27 64 04 - Código Postal 31007
SAN SEBASTIAN: Plaza de Bilbao, 1 - Tel. 42 62 37 - Télex 38095-IART - Código Postal 20005

¡¡PRECIOS INCREIBLES!!

Commodore-64



ZX Spectrum QL. Sinclair



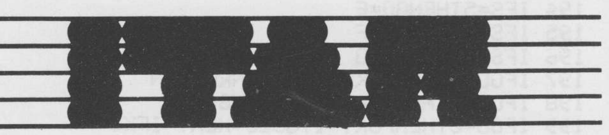
COMMODORE - 64	**Consultar**

— Unidad de discos - 1541 (170K)	**71.250 ptas.**
— Unidad de cassette	**8.900 ptas.**
— Joystick Crackshot	**2.500 ptas.**
— Juegos, utilidades, libros nacionales y extranjeros	

ZX - Spectrum (48 K)	**Consultar**
QL. - Sinclair	**Consultar**

— ZX - Interface nº 1	**15.900 ptas.**
ZX - Microdrive	**15.900 ptas.**
— ZX - Interface nº 2	**5.850 ptas.**
— Interface Joystick Kempston	**3.000 ptas.**
— Joystick Crackshot	**2.500 ptas.**
— Cartucho para ZX-Microdrive	**1.560 ptas.**
— Más de 250 juegos distintos.	
— Programas de utilidades.	
— Libros nacionales.	
— Libros extranjeros.	

SOLICITE INFORMACION



COMPUTERS, S.A.

C/ Alfonso el Batallador, 16, trasera, PAMPLONA.

BOLETIN DE PEDIDO

Nombre y apellidos ....
Dirección y teléfono ....
Deseo recibir más información ....
Deseo adquirir ....
Tarjeta VISA número ....
Fecha caducidad ....
Firma:

VIC-20

```
5 POKE36879,104:PRINT"[CLR][WHT][2CRSRD]
[4CRSRR][3SPC]MORSE"
6 PRINT"[CRSRD][2CRSRR][CRSRD][RVSON]
P[RVSOFF]RACTICA,[SPC][RVSON]E[RVSOFF]
XAMEN[CRSRD]":POKE36878,15
7 PRINT"[CRSRD][5CRSRR]O[SPC][RVSON]
T[RVSOFF]EXTO?"
8 GETK$:IFK$=""THEN8
9 IFK$="P"THEN15
12 IFK$="E"THEN52
13 IFK$="T"THENCLR:RESTORE:DIMS$(120
):K$="T":GOTO285
14 K$="P"
15 PRINT"[CLR][CRSRD][2CRSRR]MODO[SPC]
DE[SPC]PRACTICA":PRINT"[2CRSRD][CRSRR]
[RVSON]F1[RVSOFF]:[SPC]PARAR":PRINT"
[CRSRD][CRSRR][RVSON]F3[RVSOFF]:[SPC]
CAMBIAR[SPC]VEL."
16 PRINT"[CRSRD][CRSRR][RVSON]F5[RVSOFF]
:[SPC]VOLVER[SPC]AL[SPC]MENU":GOSUB6
0:GOTO84
52 PRINT"[CLR][CRSRD][2CRSRR]MODO[SPC]
DE[SPC]EXAMEN"
54 PRINT"[2CRSRD][CRSRR][CRSRD]INTRO
DUCE[SPC]UNA[SPC]LETRA"
55 PRINT"[CRSRD][SPC]O[SPC]UN[SPC]NU
MERO[SPC]TRAS":PRINT"[CRSRD][CRSRR]L
AS[SPC]SERIES[SPC]DE[SPC]SONIDOS"
56 PRINT"[3CRSRD][2CRSRR]PULSA[SPC]U
NA[SPC]TECLA"
57 POKE198,0:WAIT198,1:K=210:Y=0:Z=0
:POKE198,0:PRINT"[CLR]":GOSUB60:GOTO
85
60 PRINT"[HOM]"SPC(220)"VEL.[SPC]?[CRSRD]
[2CRSRL]A)28[SPC]LETRAS/MIN."
61 PRINT"[4CRSRR]B)34[CRSRD][4CRSRL]
C)40[CRSRD][4CRSRL]D)48"
62 PRINT"[4CRSRR]E)66[CRSRD][4CRSRL]
F)80[CRSRD][4CRSRL]G)104"
65 GETC$:IFC$="A"THENH=600:POKE7925,
90:RETURN
67 IFC$="B"THENH=500:POKE7947,90:RET
URN
70 IFC$="C"THENH=400:POKE7969,90:RET
URN
72 IFC$="D"THENH=300:POKE7991,90:RET
URN
75 IFC$="E"THENH=200:POKE8013,90:RET
URN
77 IFC$="F"THENH=150:POKE8035,90:RET
URN
80 IFC$="G"THENH=100:POKE8057,90:RET
URN
81 IFC$="[F1]"THENEND
82 IFC$="[F5]"THEN5
83 GOTO65
84 POKE198,0:I=H/3:X=INT(RND(1)*43)+
48:A$=CHR$(X)
85 IFX=58ORX=59ORX=60ORX=61ORX=62ORX
=64THEN84
86 IFK$="P"THENGETA$:PRINT"[HOM]";A$
:IFA$=""THEN86
90 IFA$="A"THENA=I:B=H:C=0:GOTO190
92 IFA$="B"THENA=H:B=I:C=I:D=I:E=0:G
OTO190
93 IFA$="C"THENA=H:B=I:C=H:D=I:E=0:G
OTO190
95 IFA$="D"THENA=H:B=I:C=I:D=0:GOTO1
90
96 IFA$="E"THENA=I:B=0:GOTO190
97 IFA$="F"THENA=I:B=I:C=H:D=I:E=0:G
OTO190
100 IFA$="G"THENA=H:B=H:C=I:D=0:GOTO
190
102 IFA$="H"THENA=I:B=I:C=I:D=I:E=0:
GOTO190
105 IFA$="I"THENA=I:B=I:C=0:GOTO190
106 IFA$="J"THENA=I:B=H:C=H:D=H:E=0:
GOTO190
107 IFA$="K"THENA=H:B=I:C=H:D=0:GOTO
190
110 IFA$="L"THENA=I:B=H:C=I:D=I:E=0:
GOTO190
112 IFA$="N"THENA=H:B=I:C=0:GOTO190
115 IFA$="M"THENA=H:B=H:C=0:GOTO190
116 IFA$="O"THENA=H:B=H:C=H:D=0:GOTO
190
117 IFA$="P"THENA=I:B=H:C=H:D=I:GOTO
190
120 IFA$="Q"THENA=H:B=H:C=I:D=H:E=0:
GOTO190
123 IFA$="R"THENA=I:B=H:C=I:D=0:GOTO
190
125 IFA$="S"THENA=I:B=I:C=I:D=0:GOTO
190
127 IFA$="T"THENA=H:B=0:GOTO190
130 IFA$="U"THENA=I:B=I:C=H:D=0:GOTO
190
132 IFA$="V"THENA=I:B=I:C=I:D=H:E=0:
GOTO190
135 IFA$="X"THENA=H:B=I:C=I:D=H:E=0:
GOTO190
136 IFA$="W"THENA=I:B=H:C=H:D=0:GOTO
190
137 IFA$="Y"THENA=H:B=I:C=H:D=H:E=0:
GOTO190
140 IFA$="Z"THENA=H:B=H:C=I:D=I:E=0:
GOTO190
142 IFA$="0"THENA=H:B=H:C=H:D=H:E=H:
F=0:GOTO190
145 IFA$="1"THENA=I:B=H:C=H:D=H:E=H:
F=0:GOTO190
147 IFA$="2"THENA=I:B=I:C=H:D=H:E=H:
F=0:GOTO190
150 IFA$="3"THENA=I:B=I:C=I:D=H:E=H:
F=0:GOTO190
152 IFA$="4"THENA=I:B=I:C=I:D=I:E=H:
F=0:GOTO190
155 IFA$="5"THENA=I:B=I:C=I:D=I:E=I:
F=0:GOTO190
157 IFA$="6"THENA=H:B=I:C=I:D=I:E=10
0:F=0:GOTO190
160 IFA$="7"THENA=H:B=H:C=I:D=I:E=I:
F=0:GOTO190
163 IFA$="8"THENA=H:B=H:C=H:D=I:E=I:
F=0:GOTO190
165 IFA$="9"THENA=H:B=H:C=H:D=H:E=I:
F=0:GOTO190
170 IFA$="."THENA=I:B=H:C=I:D=H:E=I:
F=H:G=0:GOTO190
171 IFA$="?"THENA=I:B=I:C=H:D=H:E=I:
F=I:G=0:GOTO190
172 IFA$="-"THENA=H:B=I:C=I:D=I:E=H:
F=0:GOTO190
176 IFK$="T"THENIFA$="[SPC]"THENFORT
=1TO2*I:NEXT
177 IFA$="[F1]"THENIFK$="P"THENEND
178 IFA$="[F3]"THENIFK$="P"THEN15
179 IFA$="[F5]"THENIFK$="P"THENCLR:G
OTO5
180 IFK$="E"THEN84
182 IFK$="T"THENRETURN
185 GOTO86
190 FORS=1TO7:IFS=1THENGG=A
191 IFS=2THENGG=B
192 IFS=3THENGG=C
193 IFS=4THENGG=D
194 IFS=5THENGG=E
195 IFS=6THENGG=F
196 IFS=7THENGG=G
197 IFGG=0THENIFK$="T"THENRETURN
198 IFGG=0THENIFK$="P"THEN86
199 IFGG=0THENFORT=1TO325:NEXT:IFK$=
"E"THEN255
200 POKE36876,210:FORT=1TOGG:NEXT:PO
KE36876,0:NEXT
210 IFK$="E"THEN84
```

```
211 GOTO86
255 GETB$:IFB$=""THEN255
260 IFB$=A$THENPRINT"[HOM][CRSRD][CRSRR]
BIEN[SPC]!!("A$")[4SPC]":Y=Y+1:Z=Z+1

265 IFB$><A$THENPRINT"[HOM][CRSRD][CRSRR]
MAL[2SPC]!!("A$")":Z=Z+1
270 PRINT"[CRSRD]"Y"BIEN[SPC]DE[SPC]
"Z:IFZ=10THENPRINT"[2CRSRD]"Y*10"%[SPC]
CORRECTAS":GOTO280
275 GOTO84
280 PRINT"[CRSRD][CRSRR]OTRA[SPC]VEZ
??"
281 GETD$:IFD$=""THEN281
282 IFD$="S"THENY=0:Z=0:GOTO5
283 IFD$<>"S"THENEND
284 GOTO281
285 GOSUB60:PRINT"[CLR][CRSRD]↑[CRSRD]
"
287 PRINT"[CRSRD][SPC]ESCRIBE[SPC]EL
[SPC]TEXTO[SPC]Y[SPC]PULSA.":PRINT"[CRSRD]
[SPC]RETURN.CUANDO[SPC]LO[SPC]DESCIF
RE"
288 PRINT"PULSA[SPC][RVSON]F7[RVSOFF]
[SPC]PARA":PRINT"COMPROBARLO.";CHR$(
13)
290 I=H/3:FORY=1TO44
302 GETS$(Y):IFS$(Y)=""THEN302
304 U=Y:PRINT"[HOM]"U:IFASC(S$(Y))=1
3THENY=44
306 NEXTY
307 FORY=1TOU:A$=S$(Y):FORT=1TOI:NEX
T:GOSUB90:NEXT
308 GETF$:IFF$=""THEN308
309 IFF$="[F7]"THEN312
310 GOTO308
312 PRINTSPC(220):FORY=1TOU:A$=S$(Y)
:PRINTA$;:FORT=1TOI:NEXT:GOSUB90:NEX
T
320 POKE198,0:GOTO280
```

C-64

```
5 POKE53280,0:POKE53281,6:PRINT"[CLR]
[WHT][2CRSRD][8CRSRR][5SPC]MORSE[6SPC]
"
6 PRINT"[CRSRD][9CRSRR][CRSRD][RVSON]
P[RVSOFF]RACTICA,[SPC][RVSON]E[RVSOFF]
XAMEN[CRSRD]":POKE54296,15:SS=54272
7 PRINT"[CRSRD][11CRSRR]O[SPC][RVSON]
T[RVSOFF]EXTO?"
8 GETK$:IFK$=""THEN8
9 IFK$="P"THEN15
12 IFK$="E"THEN52
13 IFK$="T"THENCLR:RESTORE:DIMS$(120
):K$="T":GOTO285
14 K$="P"
15 PRINT"[CLR][CRSRD][2CRSRR]MODO[SPC]
DE[SPC]PRACTICA":PRINT"[2CRSRD][CRSRR]
[RVSON]F1[RVSOFF]:[SPC]PARAR":PRINT"
[CRSRD][CRSRR][RVSON]F3[RVSOFF]:[SPC]
CAMBIAR[SPC]VELOCIDAD"
16 PRINT"[CRSRD][CRSRR][RVSON]F5[RVSOFF]
:[SPC]VOLVER[SPC]AL[SPC]MENU":GOSUB6
0:GOTO84
52 PRINT"[CLR][CRSRD][2CRSRR]MODO[SPC]
DE[SPC]EXAMEN"
54 PRINT"[2CRSRD][CRSRR][CRSRD]INTRO
DUCE[SPC]UNA[SPC]LETRA[SPC](O"
55 PRINT"[2CRSRD][SPC]UN[SPC]NUMERO)
[SPC]TRAS[SPC]CADA":PRINT"[CRSRD][CRSRR]
SERIE[SPC]DE[SPC]SONIDOS"
56 PRINT"[3CRSRD][2CRSRR]PULSA[SPC]U
NA[SPC]TECLA"
57 POKE198,0:WAIT198,1:K=210:Y=0:Z=0
:POKE198,0:PRINT"[CLR]":GOSUB60:GOTO
85
```

```
60 PRINT"[HOM]"SPC(255)SPC(145)"VELO
CIDAD?[CRSRD][4CRSRL]A)28[SPC]LETRAS
/MIN."
61 PRINT"[6CRSRR]B)34[CRSRD][4CRSRL]
C)40[CRSRD][4CRSRL]D)48"
62 PRINT"[6CRSRR]E)66[CRSRD][4CRSRL]
F)80[CRSRD][4CRSRL]G)104"
65 GETC$:IFC$="A"THENH=600:POKE1467,
90:RETURN
67 IFC$="B"THENH=500:POKE1507,90:RET
URN
70 IFC$="C"THENH=400:POKE1547,90:RET
URN
72 IFC$="D"THENH=300:POKE1587,90:RET
URN
75 IFC$="E"THENH=200:POKE1627,90:RET
URN
77 IFC$="F"THENH=150:POKE1667,90:RET
URN
80 IFC$="G"THENH=100:POKE1707,90:RET
URN
81 IFC$="[F1]"THENEND
82 IFC$="[F5]"THEN5
83 GOTO65
84 POKE198,0:I=H/3:X=INT(RND(1)*43)+
48:A$=CHR$(X)
85 IFX=58ORX=59ORX=60ORX=61ORX=62ORX
=64THEN84
86 IFK$="P"THENGETA$:PRINT"[HOM]";A$
:IFA$=""THEN86
90 IFA$="A"THENA=I:B=H:C=0:GOTO190
92 IFA$="B"THENA=H:B=I:C=I:D=I:E=0:G
OTO190
93 IFA$="C"THENA=H:B=I:C=H:D=I:E=0:G
OTO190
95 IFA$="D"THENA=H:B=I:C=I:D=0:GOTO1
90
96 IFA$="E"THENA=I:B=0:GOTO190
97 IFA$="F"THENA=I:B=I:C=H:D=I:E=0:G
OTO190
100 IFA$="G"THENA=H:B=H:C=I:D=0:GOTO
190
102 IFA$="H"THENA=I:B=I:C=I:D=I:E=0:
GOTO190
105 IFA$="I"THENA=I:B=I:C=0:GOTO190
106 IFA$="J"THENA=I:B=H:C=H:D=H:E=0:
GOTO190
107 IFA$="K"THENA=H:B=I:C=H:D=0:GOTO
190
110 IFA$="L"THENA=I:B=H:C=I:D=I:E=0:
GOTO190
112 IFA$="N"THENA=H:B=I:C=0:GOTO190
115 IFA$="M"THENA=H:B=H:C=0:GOTO190
116 IFA$="O"THENA=H:B=H:C=H:D=0:GOTO
190
117 IFA$="P"THENA=I:B=H:C=H:D=I:GOTO
190
120 IFA$="Q"THENA=H:B=H:C=I:D=H:E=0:
GOTO190
123 IFA$="R"THENA=I:B=H:C=I:D=0:GOTO
190
125 IFA$="S"THENA=I:B=I:C=I:D=0:GOTO
190
127 IFA$="T"THENA=H:B=0:GOTO190
130 IFA$="U"THENA=I:B=I:C=H:D=0:GOTO
190
132 IFA$="V"THENA=I:B=I:C=I:D=H:E=0:
GOTO190
135 IFA$="X"THENA=H:B=I:C=I:D=H:E=0:
GOTO190
136 IFA$="W"THENA=I:B=H:C=H:D=0:GOTO
190
137 IFA$="Y"THENA=H:B=I:C=H:D=H:E=0:
GOTO190
140 IFA$="Z"THENA=H:B=H:C=I:D=I:E=0:
GOTO190
142 IFA$="0"THENA=H:B=H:C=H:D=H:E=H:
F=0:GOTO190
145 IFA$="1"THENA=I:B=H:C=H:D=H:E=H:
F=0:GOTO190
```

```
147 IFA$="2"THENA=I:B=I:C=H:D=H:E=H:
F=0:GOTO190
150 IFA$="3"THENA=I:B=I:C=I:D=H:E=H:
F=0:GOTO190
152 IFA$="4"THENA=I:B=I:C=I:D=I:E=H:
F=0:GOTO190
155 IFA$="5"THENA=I:B=I:C=I:D=I:E=I:
F=0:GOTO190
157 IFA$="6"THENA=H:B=I:C=I:D=I:E=10
0:F=0:GOTO190
160 IFA$="7"THENA=H:B=H:C=I:D=I:E=I:
F=0:GOTO190
163 IFA$="8"THENA=H:B=H:C=H:D=I:E=I:
F=0:GOTO190
165 IFA$="9"THENA=H:B=H:C=H:D=H:E=I:
F=0:GOTO190
170 IFA$="."THENA=I:B=H:C=I:D=H:E=I:
F=H:G=0:GOTO190
171 IFA$="?"THENA=I:B=I:C=H:D=H:E=I:
F=I:G=0:GOTO190
172 IFA$="-"THENA=H:B=I:C=I:D=I:E=H:
F=0:GOTO190
176 IFK$="T"THENIFA$="[SPC]"THENFORT
=1TO2*I:NEXT
177 IFA$="[F1]"THENIFK$="P"THENEND
178 IFA$="[F3]"THENIFK$="P"THEN15
179 IFA$="[F5]"THENIFK$="P"THENCLR:G
OTO5
180 IFK$="E"THEN84
182 IFK$="T"THENRETURN
185 GOTO86
190 FORS=1TO7:IFS=1THENGG=A
191 IFS=2THENGG=B
192 IFS=3THENGG=C
193 IFS=4THENGG=D
194 IFS=5THENGG=E
195 IFS=6THENGG=F
196 IFS=7THENGG=G
197 IFGG=0THENIFK$="T"THENRETURN
198 IFGG=0THENIFK$="P"THEN86
199 IFGG=0THENFORT=1TO325:NEXT:IFK$=
"E"THEN255
200 POKE54278,100:POKE54273,34:POKE5
4272,75:POKE54276,33
201 FORT=1TOGG:NEXT
202 POKE54278,0:POKE54276,0
203 NEXT
210 IFK$="E"THEN84
211 GOTO86
255 GETB$:IFB$=""THEN255
260 IFB$=A$THENPRINT"[HOM][CRSRD][CRSRR]
BIEN!!("A$")[4SPC]":Y=Y+1:Z=Z+1
265 IFB$><A$THENPRINT"[HOM][CRSRD][CRSRR]
MAL!!!!("A$")":Z=Z+1
270 PRINT"[CRSRD]"Y"CORRECTO[SPC]DE"
Z:IFZ=10THENPRINT"[2CRSRD]"Y*10"%[SPC]
BIEN":GOTO280
275 GOTO84
280 PRINT"[CRSRD][CRSRR]OTRA[SPC]VEZ
?"
281 GETD$:IFD$=""THEN281
282 IFD$="S"THENY=0:Z=0:GOTO5
283 IFD$<>"S"THENEND
284 GOTO281
285 GOSUB60:PRINT"[CLR][11CRSRR]MODO
[SPC]TEXTO":PRINT"[HOM][CRSRD]↑[CRSRD]
"
287 PRINT"[CRSRD][SPC]ESCRBE[SPC]EL[SPC]
TEXTO[SPC]Y[SPC]PULSA[SPC]RETURN.":P
RINT"[SPC](120[SPC]CARACTERES[SPC]CO
MO[SPC]MAXIMO)
288 PRINT"[CRSRD][SPC]DESPUES[SPC]DE
[SPC]DESCIFRADO"
289 PRINT"[SPC]EL[SPC]CODIGO,[SPC]PU
LSA[SPC][RVSON]F7[RVSOFF][SPC]PARA":
PRINT"[SPC]COMPROBARLO.";CHR$(13)
290 I=H/3:FORY=1TO120
302 GETS$(Y):IFS$(Y)=""THEN302
304 U=Y:PRINT"[HOM]"U:IFASC(S$(Y))=1
3THENY=120:GOTO308
305 NEXT
306 GETH$:IFH$=""THEN306
307 IFH$<>CHR$(13)THEN306
308 FORY=1TOU:A$=S$(Y):FORT=1TOI:NEX
T:GOSUB90:NEXT
309 GETF$:IFF$=""THEN309
310 IFF$="[F7]"THEN312
311 GOTO309
312 PRINTSPC(255)SPC(145):FORY=1TOU:
A$=S$(Y):PRINTA$;:FORT=1TOI:NEXT:GOS
UB90
320 NEXT:POKE198,0:GOTO280
```

Número fantasma

Julián Alvarez García
C/Fernández de la Puente, 11 - 9ºB
Badajoz

VIC-20
sin ampliación



Este juego es para dos jugadores. Consiste en acertar un número, de 1 al 100 que el ordenador obtiene aleatoriamente por un RND. Los jugadores iran diciendo números para intentar averiguarlos y el ordenador dirá si el número que han dicho es ALTO (mayor que el nº a acertar) o BAJO.

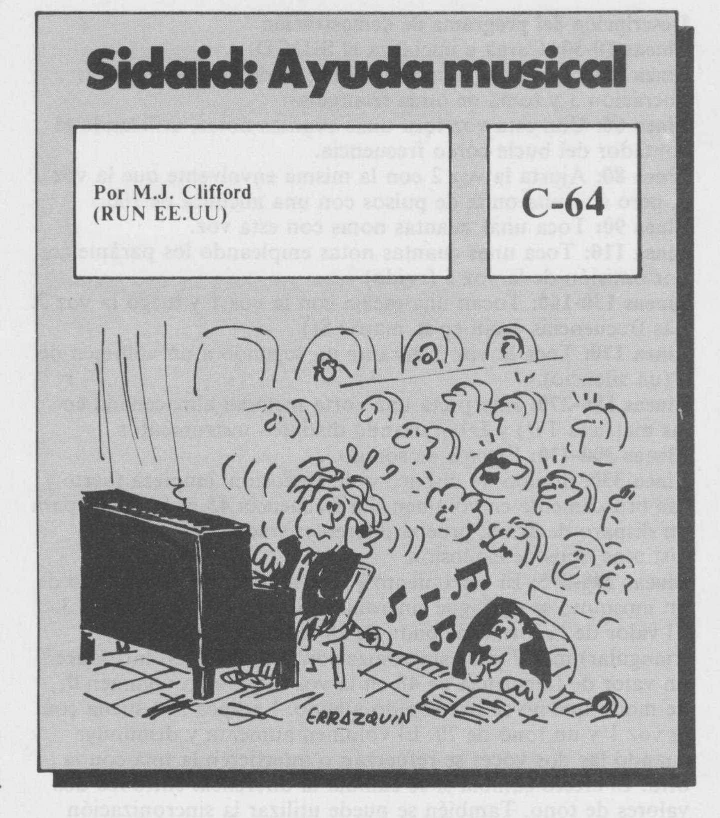
```
10 PRINT"[CLR]"
20 PRINT"[RED]COPIRYGTH[SPC]JULIAN"
30 PRINT"[BLU][8CRSRD][3CRSRR]NUMERO
[SPC]FANTASMA"
40 PRINT"[3CRSRD]NOMBRE[SPC]DEL[SPC]
PRIMER[SPC]JUGADOR[SPC]"
41 INPUT Q$
45 PRINT"[CRSRD]NOMBRE[SPC]DEL[SPC]O
TRO[SPC]JUGADOR[SPC]"
46 INPUT W$
48 PRINT"[CLR]"
49 A=(INT(RND(2)*100)+1)
50 PRINTQ$;:GOSUB1000
60 INPUT M
61 IF M<A THEN PRINT"BAJO"
62 IF M>A THEN PRINT"ALTO"
70 IF M=A THEN 250
80 PRINTW$;:GOSUB1000
90 INPUT N
91 IF N<A THEN PRINT"BAJO"
92 IF N>A THEN PRINT"ALTO"
100 IF N=A THEN 250
```

```
200 GOTO50
250 PRINT"[PUR][CRSRD][3SPC]A[SPC]C[SPC]
E[SPC]R[SPC]T[SPC]A[SPC]S[SPC]T[SPC]
E"
252 PRINT
253 PRINT"[BLK]EL[SPC]NUMERO[SPC]FAN
TASMA[SPC]ERA:[8CRSRR][RED]";A
255 IF M=A THEN K=K+1
256 IF N=A THEN O=O+1:PRINT""
257 PRINT"[CRSRD][RED]"Q$;":";K
258 PRINTW$;":";O
259 GOSUB2000
260 PRINT"[CRSRD][BLU]PARA[SPC]ACABA
R[SPC]PULSE[SPC](F)[SPC]PAERA[SPC]CO
NTINUAR[7SPC]CUALQUIER[SPC]OTRA[SPC]
LETRA"
261 GETY$:IFY$=""THEN261
262 IF Y$="F" THEN PRINT"[CLR]":END
270 GOTO 48
1000 POKE36878,15:FORT=1TO10:POKE368
75,166:NEXT:POKE36875,0:RETURN
2000 POKE36878,15
2010 FORT=128TO177STEP4:POKE36874,T:
FORM=1TO75:NEXTM:POKE36874,0:NEXTT
2015 POKE36875,241:FORJ=1TO80:NEXTJ:
POKE36875,0
2030 RETURN
```

Sidaid: Ayuda musical

Por M.J. Clifford
(RUN EE.UU)

C-64



Trabaja mientras tú tocas. Con este programa se añaden tres comandos al Basic, lo que te permitirá componer música con tu Commodore 64 sin tener que perderte en detalles.

El Commodore 64, con su sintetizador de sonido, es capaz de producir interesantes efectos musicales y sonoros. No obstante, la falta de comandos de Basic para manejar el chip SID hace que la producción de sonido se convierta en una ardua tarea. En el chip SID se utilizan 25 registros para el sonido, y cada nota requiere dos bytes de datos de frecuencia. Tendrás que tener en cuenta todas las direcciones de registros y consultar una tabla para obtener los datos de frecuecia. Para poder hacer todo esto correctamente vas a necesitar un ordenador.

Bueno, pero además de ser un sintetizador de sonido, el 64 es un ordenador; ¿por qué no dejar que se encargue de todos estos detalles?

Con el programa que se presenta a continuación el 64 hará precisamente eso. Bastará con una dirección y el ordenador hará el resto. De este modo tú quedas en libertad para concentrarte en la música y experimentar con distintos sonidos.

Han llegado los refuerzos. El programa añade tres comandos al Basic: uno para alterar la envolvente de una voz, otro para tocar una nota durante un intervalo determinado y un tercero para apagar el volumen una vez obtenido el sonido. Para ello nos valdremos del comando SYS del Basic.

El comando SYS transfiere el control a una rutina en lenguaje máquina que lee los valores necesarios en el programa Basic. Estos valores se colocan luego en los registros apropiados o se usan para encontrar los datos de frecuencia en una tabla.

Para utilizar este programa será necesario definir en primer lugar tres variables: QU de Quiet (Silencio), AV de Alter Voice (Cambia Voz) y PY de Play (Tocar) una nota. Empecemos por hacer que QU sea 40384, PY igual a QU+3 y AV igual a QU+6. Al final, los tres comandos tendrán la forma siguiente:

SYSAV,v,a,d,s,l,f(,ap)
SYSPY,v,fr,t,vl
SYSQU

Los valores representados por las minúsculas pueden ser constantes, variables o expresiones y representan lo siguiente:

v: voz (1-3)
a, d, s, l: ataque, caída, sustento, liberación (0-15).
f: forma de onda-16=triangular; 32=diente de sierra; 64=pulsante y 128=ruido. Se puede utilizar el 20 para modulación en anillo y cualquiera de los valores puede aumentarse en 2 para sincronizar con otra voz.
ap: anchura de pulso - sólo se usa con la forma de onda 64, la onda de pulsos, y puede tener valores entre 0 y 4095.
fr: 0-86, representa medios tonos desde el LA sostenido en octava 0 a SI en la octava 7 (ver tabla). 86 representa un silencio.
t: 0-255 (0=256) representa el número aproximado de fracciones de segundo (1/60º de segundo) hasta que la nota se detiene.
vl: volumen, 0-15

Nº de Octava	DO	DO#	RE	RE#	MI	FA	FA#	SOL	SOL#	LA	LA#	SI
0											0	1
1	2	3	4	5	6	7	8	9	10	11	12	13
2	14	15	16	17	18	19	20	21	22	23	24	25
3	26	27	28	29	30	31	32	33	34	35	36	37
4	38	39	40	41	42	43	44	45	46	47	48	49
5	50	51	52	53	54	55	56	57	58	59	60	61
6	62	63	64	65	66	67	68	69	70	71	72	73
7	74	75	76	77	78	79	80	81	82	83	84	85

Notas

Tabla de valores de frecuencia en siete octavas

Ejemplos

Para hacer que la voz 3 sea como un tambor, introduzca:
SYSAV,3,0,9,0,9,128

Esto hace que la voz 3 tenga un valor de ataque de 0 y un valor de caída de 9, consiguiendo un sonido que alcanza su pleno volumen inmediatamente y cae bastante rápidamente al nivel sostenido que está fijado en 0. La liberación se fija en 9 para continuar la caída si es necesario. La onda corresponde a 128, que es ruido.

Para producir el sonido de pasos que se alejan usando la voz 3 tal y como se ha definido antes, introduzca:
FOR X=15 TO 0 STEP -1:SYSPY,3,25,30,X:NEXT X

Con estas sentencias, la voz 3 suena en el tono 25 durante 30 marcas de reloj (1/2 segundo). Puesto que a la voz 3 se le asignó el nivel sostenido 0, el sonido es el mismo para prácticamente todos los valores de tiempo. Por tanto, el 30

sirve para controlar el tiempo entre sonidos. El bucle toca el sonido 15 veces con un volumen cada vez menor, desde un máximo de 15 hasta 0, según queda determinado por el valor x.

Para hacer que la voz 1 sea un piano, introduzca:
SYSAV,1,0,9,0,0,64,255

El piano tiene una envolvente similar a la de un tambor, pero el 64 hace que la forma de onda sea un tren de pulsos con una anchura de 255.

Si se quiere evitar el zumbido del altavoz cuando no se toca ninguna nota, se puede desactivar totalmente el chip SID introduciendo:
SYSQU

Los ejemplos presentados son para usar desde un programa en el que QU, AV y PY tienen los valores correctos. Para utilizar los comandos en modo Directo, ha de utilizar las direcciones reales. Así, en modo Directo SYSQU es SYS40384; SYSAV es SYS40390; y SYSPY es SYS40387.

El programa en el Listado 1 tiene dos objetivos. Lo primero que hace es escribir el programa en lenguaje máquina en la memoria, luego lo copia en disco o cinta. Una vez ejecutado el programa, sólo lo necesitarás para guardar una copia del programa SIDAID en otro disco o cinta. Para cargar el programa SIDAID, introduce:
LOAD"SIDCD",8,1 (,1,1 para cinta)
POKE52,157:POKE 56,157:NEW

Los comandos POKE bajan el final de memoria disponible del Basic en 768 bytes para evitar que el Basic "machaque" el programa en lenguaje máquina. Una vez hecho esto, los comandos quedan disponibles en modo Directo, o se puede introducir, cargar o ejecutar un programa que los utilice.

También se puede realizar esta operación desde el propio programa que utiliza el SIDAID, para lo cual basta añadir al principio las siguientes líneas:

```
10 POKE 52,157:POKE 56,157:CLR
20 IF PEEK(40384)<>76 THEN
   LOAD"SIDCD",8,1
30 QU =40384:PY=QU+3:AV=QU+6
```

La comprobación de la línea 20 es necesaria puesto que un comando LOAD ejecutado desde un programa Basic hará que el programa empiece desde el principio una vez que esté cargado. Esta comprobación evita que se ejecute repetidamente el proceso de carga, y permite que el programa continúe una vez que SIDCD está preparado.

El listado 2 contiene un pequeño programa de demostración que ilustra algunas de las posibilidades del paquete. Se pueden hacer muchas otras cosas además de las que hemos visto: este paquete facilita extraordinariamente la experimentación. Dos de las características demostradas en este programa, sincronización y modulación en anillo, merecen una explicación más detallada.

La sincronización es el uso del comando AND para combinar dos formas de onda, lo que hace que el volumen fluctúe cuando las dos ondas se refuerzan o interfieren una con otra. Esta característica se activa añadiendo 2 al valor f (forma de onda) del comando SYSAV.

Cuando se usa la sincronización con la voz 1, la voz 3 se ha de ajustar a una frecuencia distinta de cero y, preferiblemente, menor que la frecuencia de la voz 1. Todos los otros parámetros de la voz 3 pierden efecto. Cuando se activa la sincronización para la voz 2, se sincroniza con la frecuencia de la voz 1. La voz 3 se sincroniza con la voz 2.

La modulación en anillo se activa utilizando un valor f (forma de onda) de 20 en el comando SYSAV, con ello se obtienen sonidos de campana o de gong. La modulación en anillo de la voz 1 requiere que la voz 3 esté en una frecuencia algo más baja. La voz 2 se modula con la voz 1 y la voz 3 se modula con la frecuencia de la voz 2. La calidad del sonido depende en parte de la diferencia de frecuencia entre la voz que suena y la voz modulada.

Descripción del cargador

Líneas 10-20: Protegen el programa en lenguaje máquina contra el Basic y establece la dirección inicial.

Líneas 30-70: Leen los datos de las líneas 1001-1037, saltando la línea 1026, que es un REM. El último número de cada línea de datos es un dato de comprobación. Si se equivoca al introducir algún dato, el programa se detiene e indica el número de la línea que contiene el dato erróneo.

Líneas 80-100: Avisan de que el programa se ha cargado satisfactoriamente.

Líneas 200-280: Utilizan la rutina de Almacenamiento de Kernal para guardar el programa en lenguaje máquina en un fichero de programas que se puede cargar directamente con LOAD"SIDCD",8,1 desde disco, o LOAD"SIDCD",1,1 desde cassette.

Línea 200: Mete en F$ el nombre del programa.

Línea 210-220: Busca la posición de F$ en la memoria.

Líneas 230-240: Escribe (con Poke) la longitud y la dirección de F$ donde la rutina de Kernal supone que está.

Línea 250: Escribe (con Poke) la dirección inicial en la memoria disponible de la página cero.

Línea 260: Escribe el número del dispositivo periférico y la dirección secundaria de forma que estos dos datos estén a disposición de la Rutina de Kernal. Cambia el 8 por 1 si vas a usar una cinta.

Línea 270: Escribe la posición de la dirección inicial y los bytes alto y bajo de la dirección final. Estos datos también quedan a disposición de la Rutina de Kernal.

Línea 280: Llama a la rutina de almacenamiento.

Líneas 290-320: Indican que se ha llevado a cabo el almacenamiento. Cuando uses cassettes, cambie el 8 de la línea 320 por un 1.

Descripción del programa de demostración

Líneas 10-30: Carga e inicializa el SIDCD.

Línea 50: Fija la voz 1 con ataque 0, caída 0, sostenido 15, liberación 3 y toma de onda triangular.

Línea 60: Con esta voz toca unas cuantas notas, utilizando el contador del bucle como frecuencia.

Línea 80: Ajusta la voz 2 con la misma envolvente que la voz 1, pero con una onda de pulsos con una anchura de 100.

Línea 90: Toca unas cuantas notas con esta voz.

Línea 110: Toca unas cuantas notas empleando los parámetros por omisión de la voz 3 (ruido).

Líneas 130-160: Tocan una escala con la voz 1 y luego la voz 2. Las frecuencias están en la matriz S().

Línea 170: Toca la voz 3 durante un segundo a un volumen de 0 (un silencio).

Líneas 180-270: Interpreta una corta melodía almacenada en las matrices T1() y D1(), usando distintos instrumentos.

Líneas 300-410: Efectos de sonido.

Línea 330: Sonido de disparo o de explosión: empieza fuerte y cae bruscamente en volumen. La frecuencia 45 es la mejor para un disparo de arma; prueba diferentes frecuencias para distintos tipos de explosión.

Líneas 340-375: El crecimiento y decrecimiento del zumbido de un mosquito se consigue sincronizando la voz 1 con la voz 3. El valor de la forma de onda de 18 equivale a 16 (para triangular) más 2 para sincronización. La línea 360 introduce un valor de frecuencia de 48 en la voz 3, pero en volumen 0, de modo que no se oye sonido alguno. La línea 370 suena con la voz 1 y un tono de 70. El volumen aumenta y disminuye cuando las dos voces se refuerzan o interfieren la una con la otra. El efecto cambia si se cambia la diferencia entre los dos valores de tono. También se puede utilizar la sincronización con las otras formas de onda.

Líneas 380-410: El sonido de campana se produce por la modulación en anillo de la forma de onda triangular que utiliza un valor 20, formado por 16 para el triángulo y 4 para el anillo. En la voz 3 se introduce un tono, como en la sincronización, y el efecto producido depende de la diferencia entre los valores de tono de las dos voces.

Líneas 500-540: Subrutina para interpretar una melodía, con los valores de tono y duración en las matrices T1() y D1(). Los valores de duración se multiplican por el valor del tempo T, de modo que la duración de la nota se puede cambiar con sólo cambiar T, como se hizo en las líneas 190-220.

Líneas 1000-1030: Inicialización o arranque del programa.

Línea 1010: Lee los datos de la línea 1040 dejándolos en la matriz S(). Los valores dados corresponden a una escala en clave de DO, empezando en el DO central. El 86 es el valor del silencio.
Línea 1020: Lee los valores de tono y duración de una corta melodía, y los almacena en las matrices T1() y D1().
Línea 1030: Inicializa el chip SID escribiendo 0 en todos los registros.

Listado I

```
5 REM CARGADOR-SIDAID M J CLIFFORD
10 POKE52,157:POKE56,157:POKE51,192:
POKE55,192:CLR
20 AD=40384
30 FORL=1TO37:IFL=26THENL=27
40 FORX=0TO15:READB:C=C+B:POKEAD+X,B
:NEXT
50 READK:IFC<>KTHENPRINT"ERROR[SPC]E
N[SPC]LA[SPC]LINEA[SPC]DATA";1000+L:
END
60 T=T+C:C=0:AD=AD+16:NEXTL
70 READK:IFT<>KTHENPRINT"ERROR[SPC]E
N[SPC]LAS[SPC]LINEAS[SPC]DATA":END
80 QU=40384:PY=QU+3:AV=QU+16
90 SYSPY,2,30,30,15:SYSQU
100 PRINT"EL[SPC]UTILITARIO[SPC]SIDA
ID[SPC]ESTA[SPC]CARGADO"
110 PRINT"SALVANDO[SPC]EL[SPC]PROGRA
MA[SPC]DE[SPC]C.M.[CRSRD]"
200 F$="SIDCD"
210 POKE187,PEEK(71):POKE188,PEEK(72
)
220 FA=PEEK(187)+256*PEEK(188)
230 POKE183,PEEK(FA)
240 POKE187,PEEK(FA+1):POKE188,PEEK(
FA+2)
250 POKE251,192:POKE252,157
260 POKE186,8:POKE185,1
270 POKE780,251:POKE781,0:POKE782,16
0
280 SYS65496
290 SYSPY,2,50,30,15:SYSQU
300 PRINT"EL[SPC]UTILITARIO[SPC]SIDA
ID[SPC]ESTA[SPC]GUARDADO[SPC]COMO[SPC]
"F$
310 PRINT"LA[SPC]PROXIMA[SPC]VEZ[SPC]
TECLEA[SPC]SOLO:"
320 PRINT"LOAD"CHR$(34)F$CHR$(34)",8
,1[2CRSRD]"
1000 REM ***DATOS DEL SIDCD***
1001 DATA76,45,159,76,78,158,32,69,1
58,152,41,3,240,97,141,50,1575
1002 DATA158,206,50,158,32,69,158,15
2,10,10,10,10,133,251,32,69,1508
1003 DATA158,152,41,15,5,251,133,251
,32,69,158,10,10,10,10,133,1438
1004 DATA253,32,69,158,152,41,15,5,2
53,133,253,32,69,158,152,41,1816
1005 DATA246,172,50,158,153,57,158,9
,1,153,66,158,165,253,153,54,2006
1006 DATA158,165,251,153,51,158,185,
57,158,41,64,240,17,32,69,158,1957
1007 DATA72,173,50,158,10,170,104,15
7,61,158,152,157,60,158,96,108,1844
1008 DATA0,3,0,9,10,9,3,3,9,64,64,12
8,0,4,0,4,310
1009 DATA0,0,65,65,129,32,253,174,32
,158,173,76,170,177,32,69,1605
1010 DATA158,152,41,3,240,33,141,50,
158,206,50,158,32,69,158,132,1781
1011 DATA251,32,69,158,132,253,32,69
,158,140,24,212,173,50,158,240,2151
1012 DATA11,74,176,62,76,232,158,162
,14,108,0,3,173,51,158,141,1599
1013 DATA5,212,173,54,158,141,6,212,
173,60,158,141,2,212,173,61,1941
1014 DATA158,141,3,212,166,251,189,1
67,159,188,80,159,141,0,212,140,2366
1015 DATA1,212,173,66,158,141,4,212,
32,30,159,173,57,158,141,4,1721
1016 DATA212,96,173,52,158,172,55,15
8,141,12,212,140,13,212,173,62,2041
1017 DATA158,141,9,212,173,63,158,14
1,10,212,166,251,189,167,159,188,239
7
1018 DATA80,159,141,7,212,140,8,212,
173,67,158,141,11,212,32,30,1783
1019 DATA159,173,58,158,141,11,212,9
6,173,53,158,172,56,158,141,19,1938
1020 DATA212,140,20,212,173,64,158,1
41,16,212,173,65,158,141,17,212,2114

1021 DATA166,251,189,167,159,188,80,
159,141,14,212,140,15,212,173,68,233
4
1022 DATA158,141,18,212,32,30,159,17
3,59,158,141,18,212,96,160,0,1767
1023 DATA162,10,136,208,253,202,208,
250,198,253,208,244,96,169,0,141,273
8
1024 DATA24,212,141,4,212,141,11,212
,141,18,212,96,0,141,24,212,1801
1025 DATA141,4,212,141,11,212,141,18
,212,96,255,255,255,255,255,255,2718

1026 REM*** DATOS DE FRECUENCIAS ALT
O/BAJO***
1027 DATA1,1,2,2,2,2,2,2,2,3,3,3,3,3
,4,4,39
1028 DATA4,4,5,5,5,6,6,6,7,7,8,8,9,9
,10,10,109
1029 DATA11,12,12,13,14,15,16,17,18,
19,20,21,22,24,25,27,286
1030 DATA28,30,32,34,36,38,40,43,45,
48,51,54,57,61,64,68,729
1031 DATA72,76,81,86,91,96,102,108,1
15,122,129,137,145,153,163,172,1848
1032 DATA183,193,205,217,230,244,0,2
05,233,6,37,69,104,140,179,220,2465
1033 DATA8,54,103,155,210,12,73,139,
208,25,103,185,16,108,206,53,1658
1034 DATA163,23,147,21,159,50,205,11
4,32,216,156,107,70,47,37,42,1589
1035 DATA63,100,154,227,63,177,56,21
4,141,94,75,85,126,200,52,198,2025
1036 DATA127,97,111,172,126,188,149,
169,252,161,105,140,254,194,223,88,2
556
1037 DATA52,120,43,83,247,31,210,25,
252,133,189,176,103,0,0,0,1664,62217
```

Listado II

```
5 REM DEMOSTRACION DEL SIDAID
10 POKE52,157:POKE56,157:CLR
20 IFPEEK(40384)<>76THENLOAD"SIDCD",
8,1
30 QU=40384:PY=QU+3:AV=QU+6
40 GOSUB1000:REM INICIALIZACION
50 SYSAV,1,0,10,15,3,16
60 FORX=9TO79STEP2:SYSPY,1,X,5,15:NE
XT
70 SYSQU
80 SYSAV,2,0,10,15,3,64,100
90 FORX=20TO60:SYSPY,2,X,5,15:NEXT
100 SYSQU
110 FORX=35TO50:SYSPY,3,X,10,15:NEXT

120 SYSQU
130 FORV=1TO2
140 FORX=1TO8:SYSPY,V,S(X),25,15:NEX
T
150 FORX=8TO1STEP-1:SYSPY,V,S(X),15,
15:NEXT
```

```
160 NEXTV:SYSQU
170 SYSPY,3,86,60,0
180 REM TOCA UNA MELODIA CON DIFEREN
TES INSTRUMENTOS
190 T=6
200 SYSAV,1,0,9,0,0,64,255:REM PIANO

205 GOSUB500
210 SYSAV,1,0,9,0,0,64,63:REM BANJO
220 GOSUB500:T=4
240 SYSAV,1,6,5,2,2,32:REM ACORDEON
250 GOSUB500
260 SYSAV,1,6,0,10,1,32:REM TROMPETA

270 GOSUB500
300 REM EFECTOS SONOROS
310 REM DISPAROS DE PISTOLA
320 FORZ=1TO3
330 FORX=15TO0STEP-1:SYSPY,3,45,1,X:
NEXT:NEXT:SYSQU
340 REM MOSQUITO UTILIZANDO LA SINCR
ONIZACION
350 SYSAV,1,13,11,0,0,18
360 SYSPY,3,48,10,0:REM SOLO SINCRON
IZADO
370 FORX=1TO5:SYSPY,1,70,200,15:NEXT

375 SYSQU
380 REM CAMPANADAS UTILIZANDO MODULA
CION EN ANILLO
390 SYSAV,1,0,9,0,0,20
395 SYSPY,3,50,10,0
400 FORX=1TO12:SYSPY,1,74,160,15:NEX
T
410 SYSQU
499 END
500 REM TOCA UNA MELODIA
520 FORX=1TO21:SYSPY,1,T1(X),D1(X)*T
,15:NEXT
530 SYSQU
540 RETURN
999 END
1000 DIMS(8),T1(21),D1(21)
1010 FORX=0TO8:READS(X):NEXTX:REM ES
CALA
1020 FORX=1TO21:READT1(X),D1(X):NEXT

1030 FORX=54272TO54296:POKEX,0:NEXT
1035 RETURN
1039 REM ESCALA EN DO
1040 DATA86,38,40,42,43,45,47,49,50
1049 REM SINTONIA
1050 DATA45,4,47,4,45,4,45,4,45,2,47
,2,49,12,45,4,47,4,40,4,40,4,40,4,40
,4
1055 DATA42,1,43,1,45,4,43,2,40,8,38
,12,86,12,50,12
```



Conversión

Para convertir los códigos ASCII en códigos POKE de la pantalla:

1 def fn f(a)=a-161-33*(a<225)-64*(a<192)-32*(a<160) +32*(a<96)-64*(a<64)

Esta función convierte los ASCII para que podamos hacer directamente el POKE. Es útil por ejemplo en cuanto estamos con un procesador de textos.

T.H. RUN EE.UU.

Anuncios por palabras



Fig. 1. Anuncio normal por palabras, de 7,5 × 12,7 centímetros (reducido al 85% de su tamaño real).

En Venta
Henway 1973
Perfecto estado
Llamar a Pedro
853-1212

¿Tiene algo que decir, algo que vender, o algo que anunciar? Convierte tu mensaje en un verdadero anuncio y pégalo en el tablón.

Poco se ha escrito sobre la técnica del tablón de anuncios, si bien todos sabemos que el coste es bajo y la tecnología necesaria es elemental: un rotulador y una cartulina de 7,5 × 12,7 cm. representan lo más avanzado en este campo.

Tengo un VIC 20 y una impresora, así que decidí inventarme un programa que me imprimiese un bonito cartel para exponer en el tablón de anuncios de la empresa o en el kiosko del barrio. El programa que obtuve se muestra en el listado adjunto.

El tamaño del cartel lo elegí pensando en los otros anunciantes que compartirían el espacio conmigo, una cartulina de 7,5 × 12,7 cm. con el texto impreso en letras grandes y las líneas centradas. En la figura 1 se puede apreciar un anuncio típico, que tiene siete líneas de hasta 23 caracteres cada una. Sin embargo, y como veremos más adelante, estas características se pueden cambiar fácilmente.

La impresora que yo tengo sólo tiene dos densidades de caracteres, y yo elegí el tamaño ampliado de cinco caracteres por pulgadas (CPP) para el texto del anuncio. Los caracteres del marco se imprimen con densidad normal, es decir 10 cpp. En un anuncio de cinco pulgadas de anchura (12,7 cm.) entrarán veinticinco caracteres por línea a 5 cpp. Los dos caracteres del marco en los extremos de una línea equivalen a una posición de impresión ampliada, puesto que cada carácter es la mitad de la anchura de un carácter ampliado. Además, el programa impone un mínimo de un espacio al final de la línea. Por lo tanto, caben dos caracteres menos que la longitud real de la línea.

La longitud de la línea se define con la variable LL en la línea 90 del programa. Cuando necesite un cartel de diferente anchura, basta cambiar el valor de LL para definir el nuevo número de caracteres por línea y ajustar el marco al nuevo

tamaño. La figura 2 es un anuncio que tiene cinco líneas de 18 caracteres (LL = 20).

La variable NL de la línea 90 determina la altura del anuncio. NL es el número de líneas del texto. Cada línea de texto está entre dos líneas totalmente en blanco, excepto por los caracteres del marco en cada extremo. La impresora recibe la instrucción de utilizar el modo gráfico (CHR$(8)) al cambiar de línea, de modo que los caracteres del marco queden juntos verticalmente.

La línea superior del marco se imprime como una línea del carácter definido por B$, seguido de tres líneas en blanco para dejar un cierto espacio libre sobre el texto. En la parte inferior sucede lo contrario: a las tres líneas en blanco sigue la línea de caracteres del marco.

Cómo funciona. Las líneas 1 a 17 explican el uso de las variables del programa, y no es realmente necesario teclearlas. Las líneas 80 y 90 asignan valores a cada una de las variables. La idea es que se puedan modificar fácilmente en todo el programa cambiando sólo una línea. Por ejemplo, si su impresora utiliza un código gráfico distinto de CHR$(8), podrá cambiarlo en la línea 80, y siempre será correcto en el momento oportuno. B$ es el carácter del marco: CHR$(191), similar a un tablero de ajedrez. Lo puede sustituir por cualquier carácter imprimible, en caso de que quiera cambiar el marco. NL y LL controlan el tamaño vertical y horizontal del anuncio, tal y como se describió antes.

Las líneas 100-190 aceptan líneas de texto del teclado y las asignan a las variables AD$(LI), en las que LI representa el número de cada una de las líneas del texto partiendo de 1 hasta el número asignado por NL. Por ejemplo, AD$(4) contiene el texto que se va a imprimir en la cuarta línea.

Las líneas 200-290 combinan verticalmente las líneas del marco con las líneas del texto, y utilizan las subrutinas que hay por encima de la línea 1000 para centrar y preparar las líneas horizontalmente.

Las líneas 1000 y 1010 imprimen tres líneas en blanco sobre la primera y debajo de la última línea del texto.

Las líneas 1400-1470 centran el texto en la línea de impresión y añaden el carácter del marco a cada extremo. Luego se imprime la línea.

Las líneas 2100-2130 construyen e imprimen una línea que contiene una línea entera de caracteres de marco, o una línea en blanco con el carácter de marco a cada extremo.

Cómo se usa. Teclea el programa Anuncios por Palabras, omitiendo las líneas 1 a 17 si así lo deseas. Teclea RUN. El número de líneas de texto y el número máximo de caracteres por línea aparecerá en la parte superior de la pantalla. A medida que introduces cada línea, aparecerá un signo de interrogación en la parte izquierda de la pantalla, que indica que el programa está preparado para aceptar una nueva línea. No añadas espacios al principio ni al final de la línea, ya que el programa los consideraría como parte del texto y las líneas no quedarían correctamente centradas.

Cuando hayas introducido la última línea, el signo de interrogación no aparece, y la impresora comienza a imprimir el anuncio. Cuando haya terminado, si tecleas RUN no se imprime de nuevo el mismo anuncio, sino que vuelve a ejecutar el programa desde el principio. Para sacar muchos ejemplares de un mismo anuncio, añade las siguientes dos líneas al programa:

```
270 INPUT" SHFT CLEAR   OTRA COPIA";CO$
280 IFLEFT$(CO$,1)="S"GOTO220
```

Al responder "S" o "SI" se repetirá el mismo anuncio, siempre que se hayan añadido las líneas 270 y 280 al programa original. Si utilizas papel continuo, es posible que tengas que ajustarlo un poco antes de contestar a la pregunta para evitar que la máquina imprima en una perforación.

Naturalmente, este programa se puede utilizar para otras cosas aparte de los anuncios de ventas, como por ejemplo, carteles de "prohibido entrar" en la habitación de los niños. O quizá no encuentres ninguna aplicación concreta, pero aún así vale la pena experimentar con él, si te interesa la programación Basic.

Figura 2. Anuncio más pequeño donde LL=20 y NL=5. Observe que se ha cambiado el carácter de marco. (Reducido a 85% de su tamaño real).

VENTA DE LIBROS Y ROPA
SABADO 3 DE NOV.
CALLE DEL PEZ, 2

VIC-20 y C-64

```
1 REM:    ASIGNA  7.5 * 12.5  CM.
2 REM:       PARA EL ANUNCIO
3 REM:
4 REM:    A$=STRING A IMPRIMIR
5 REM:    B$=CARACTER DEL BORDE
6 REM:    C$=CADENA DE TEXTO CENTRADA

7 REM:    G$=COMANDO DE GRAFICOS
8 REM:    N$=COMANDO DE IMPRESION NOR
MAL
9 REM:    X$=COMANDO DE IMPRESION EXT
ENDIDA
10 REM:   LI=NUMERO DE LINEA ACTUAL
11 REM:   LL=LONGITUD MAXIMA DE LINEA

12 REM:   NL=NUMERO DE LINEAS
13 REM:   SL=LONGITUD DE CADENA
14 REM
15 REM
16 REM:***************************
17 REM:
80 X$=CHR$(14):G$=CHR$(8):N$=CHR$(15
)
90 B$=CHR$(191):NL=7:LL=25
100 PRINT"[CLR][5SPC]-SE[SPC]VENDE-[CRSRD]
"
110 PRINT"[SPC]TIENES";NL;"[SPC]LINE
AS"
120 PRINT"[SPC]DE";LL-2;"[SPC]CARACT
ERES.[CRSRD]"
130 FORLI=1TONL
140 INPUTAD$(LI)
150 IFLEN(AD$(LI))<=LL-2THEN190
160 POKE36878,0:GOTO140
190 NEXTLI
200 REM * LO SACA POR IMPRESORA *
210 OPEN1,4
220 R$=B$:GOSUB2100:R$="[SPC]":GOSUB
2100
230 GOSUB1000:FORLI=1TONL
240 GOSUB2100:GOSUB1400:GOSUB2100
250 NEXTLI
260 GOSUB1000:GOSUB2100:R$=B$:GOSUB2
100
290 CLOSE1:END
1000 REM* DEJA 3 LINEAS EN BLANCO *
1010 R$="[SPC]":FORQ=1TO3:GOSUB2100:
NEXTQ:RETURN
1400 REM * PREPARA UNA LINEA *
1410 SL=LEN(AD$(LI))/2
1420 C$="":FORN=1TOINT(((LL-1)/2)-SL
)
1430 C$=C$+"[SPC]":NEXTN:C$=C$+AD$(L
I)
1440 FORN=0TO(LL-SL)/2:C$=C$+"[SPC]"
:NEXTN
1450 C$=LEFT$(C$,LL-1)
1460 A$=N$+B$+X$+C$+N$+B$+G$:PRINT#1
,A$
1470 RETURN
2100 A$=N$+B$
2110 FORQ=1TO(2*LL)-2:A$=A$+R$:NEXTQ

2120 A$=A$+B$+G$:PRINT#1,A$
2130 RETURN
```

Auto VIC

Francisco Plaza Perez
Jeus Rivero Meneses, nº 2-9 C
47014 VALLADOLID

VIC-20



Al ejecutar el programa, y tras un tiempo de espera y si no ha habido ningún error de DATAS el programa nos dice que SYS tenemos que hacer para conectarlo.

Cuando queramos autonumerar cualquier programa sólo tendremos que hacer el SYS adecuado o SYS 0, entonces nos aparecerá en la pantalla:

INICIAL, INCREMENTO?

nosotros tenemos que introducir dos cantidades separadas por una coma, estas deben de ser inferiores a 100000, si no aparecera un illegal cuantity, para desconectar el autonumerador nada más fácil teclear a la vez RUN/STOP y RESTORE.

El programa en BASIC lo único que hace es quitarte esos 281 bytes para el programa de código máquina, ajustar las direcciones absolutas del programa a la configuración que tenga en ese momento el VIC-20, y cargar el programa.

Adjunto también el desensamblado del programa con algunos comentarios de lo que hace cada parte del programa. El programa se reduciría a la mitad si se introducen las cantidades INICIAL e INCREMENTO a mano (o sea a POKES).

De $1DDB a $1DDF se guarda la cantidad inicial, que se va incrementando a medida que se van introduciendo líneas. De $1DE1 a $1DE5 se guarda la cantidad a incrementar, cada byte de estos registros sólo puede tener un valor de 0 a 9, esto desaprovecha memoria, pero ahorra el software que habría que añadir para transformar los bytes en dígitos decimales. Todas las direcciones reseñadas y las del desensamblado son con la configuración standard o con la de 3Ko., con 16Ko. ó 8Ko. serían diferentes.

```
10 REM*F.PLAZA*16-4-84*530 BYTES**
20 REM***AUTOMATICO**
30 P=PEEK(55)+PEEK(56)*256:P=P-281
40 P1=INT(P/256):P2=INT((P/256-P1)*2
56+.5)
50 POKE55,P2:POKE56,P1:POKE51,P2:POK
E52,P1
54 FORT1=0TO17:Q=0:FORT=0TO15:READA:
Q=Q+A:NEXT
56 READA:IFQ<>ATHENPRINT"[CLR][3CRSRD]
[RVSON]ERROR[SPC]EN[SPC]LINEA";100+T
1*10:END
58 NEXT:Q=0:T=0:RESTORE
60 FORT1=0TO15:READA
70 IFA>999ANDA<2000THENA=A-1000+P2:I
FA>255THENA=A-256:Q=1
80 IFA>1999ANDA<3000THENA=A-2000+P1+
Q:Q=0
90 POKET+P,A:T=T+1:IFT>280THEN300
95 NEXT:READA:GOTO60
100 DATA162,0,169,0,157,1245,2000,15
7,1251,2000,232,224,5,208,245,162,82
17
110 DATA255,232,189,1000,2001,32,210
,255,224,24,208,245,162,0,32,15,5084

120 DATA225,201,13,240,13,157,0,2,23
2,224,89,144,241,162,23,76,2042
130 DATA55,196,224,1,240,53,160,5,20
2,224,0,48,52,189,0,2,1651
140 DATA201,44,240,16,56,233,48,153,
1250,2000,136,192,0,16,233,162,4980
150 DATA14,108,0,3,160,5,202,224,0,4
8,22,189,0,2,56,233,1266
160 DATA48,153,1244,2000,136,192,0,1
6,237,48,228,142,1248,2000,142,1254,
9088
170 DATA2000,173,36,3,141,1138,2000,
173,37,3,141,1139,2000,169,1140,141,
10434
180 DATA36,3,169,2000,141,37,3,76,11
6,196,14,242,8,141,64,3,3249
190 DATA104,141,65,3,142,66,3,140,67
,3,173,66,3,201,0,208,1385
200 DATA3,32,1181,2000,172,67,3,174,
66,3,173,65,3,72,173,64,4251
210 DATA3,40,108,1138,2000,162,255,1
60,1,232,189,1245,2000,208,5,224,797
0
220 DATA3,208,246,232,189,1245,2000,
24,105,48,153,118,2,200,224,4,5001
230 DATA208,241,169,32,153,119,2,132
,198,162,4,24,189,1245,2000,125,5003

240 DATA1251,2000,201,10,144,6,56,23
3,10,56,176,1,24,157,1245,2000,7570
250 DATA202,16,233,96,0,0,0,1,0,0,1,
0,0,0,1,0,550
260 DATA13,13,80,82,73,77,46,76,73,7
8,69,65,44,73,78,67,1007
270 DATA2,69,77,69,78,84,79,63,13,0,
0,0,0,0,0,0,534
300 POKE0,76:POKE1,P2:POKE2,P1:PRINT
"[CLR][3CRSRD][RVSON]PARA[SPC]CONECT
AR[SPC]EL[SPC]AUTO[SPC]HACER[SPC]SYS
0[SPC]O[SPC]SYS"P:NEW
```

```
B*
   PC  SR AC XR YR SP
.;603E 33 00 63 00 F6
.
.,  1CE7 LDX #$00
.,  1CE9 LDA #$00
.,  1CEB STA $1DDC,X
.,  1CEE STA $1DE2,X      } Borra memorias e incremento.
.,  1CF1 INX
.,  1CF2 CPX #$05
.,  1CF4 BNE $1CEB
.,  1CF6 LDX #$FF
.,  1CF8 INX
.,  1CF9 LDA $1DE7,X
.,  1CFC JSR $FFD2        } Presenta en pantalla texto.
.,  1CFF CPX #$18
.,  1D01 BNE $1CF8
```

```
., 1D03 LDX #$00
., 1D05 JSR $E10F
., 1D08 CMP #$0D
., 1D0A BEQ $1D19
., 1D0C STA $0200,X
., 1D0F INX
., 1D10 CPX #$59
., 1D12 BCC $1D05
., 1D14 LDX #$17
., 1D16 JMP $C437
```

Introduce teclas dadas en buffer de basic ($0200).

```
., 1D19 CPX #$01
., 1D1B BEQ $1D52
., 1D1D LDY #$05
```

Si no hay nada pone a 10 los dos registros.

```
., 1D1F DEX
., 1D20 CPX #$00
., 1D22 BMI $1D58
., 1D24 LDA $0200,X
., 1D27 CMP #$2C
., 1D29 BEQ $1D3B
., 1D2B SEC
., 1D2C SBC #$30
., 1D2E STA $1DE1,Y
., 1D31 DEY
., 1D32 CPY #$00
., 1D34 BPL $1D1F
```

Introduce en la memoria de "inicial" lo contenido en el buffer de basic.

```
., 1D36 LDX #$0E
., 1D38 JMP ($0300)
., 1D3B LDY #$05
```

Si hay más de 5 cifras ir a subrutina de errores.

```
., 1D3D DEX
., 1D3E CPX #$00
., 1D40 BMI $1D58
., 1D42 LDA $0200,X
., 1D45 SEC
., 1D46 SBC #$30
., 1D48 STA $1DDB,Y
., 1D4B DEY
., 1D4C CPY #$00
., 1D4E BPL $1D3D
., 1D50 BMI $1D36
```

Introduce en la memoria de "incrementos" lo contenido en el buffer de basic.

```
., 1D52 STX $1DDF
., 1D55 STX $1DE5
., 1D58 LDA $0324
., 1D5B STA $1D71
., 1D5E LDA $0325
., 1D61 STA $1D72
., 1D64 LDA #$73
., 1D66 STA $0324
., 1D69 LDA #$1D
., 1D6B STA $0325
., 1D6E JMP $C474
```

Cambia puntero de Kernal Vector de interrupción y vuelve al basic.

```
., 1D71 ASL $08F2
., 1D74 STA $0340
., 1D77 PLA
., 1D78 STA $0341
., 1D7B STX $0342
., 1D7E STY $0343
., 1D81 LDA $0342
., 1D84 CMP #$00
., 1D86 BNE $1D8B
., 1D88 JSR $1D9C
., 1D8B LDY $0343
., 1D8E LDX $0342
., 1D91 LDA $0341
., 1D94 PHA
., 1D95 LDA $0340
., 1D98 PLP
., 1D99 JMP ($1D71)
```

Intercepta rutina 'Vector de interrupción'

```
., 1D9C LDX #$FF
., 1D9E LDY #$01
., 1DA0 INX
., 1DA1 LDA $1DDC,X
., 1DA4 BNE $1DAB
., 1DA6 CPX #$03
., 1DA8 BNE $1DA0
., 1DAA INX
., 1DAB LDA $1DDC,X
., 1DAE CLC
., 1DAF ADC #$30
., 1DB1 STA $0276,Y
., 1DB4 INY
., 1DB5 CPX #$04
., 1DB7 BNE $1DAA
```

Introduce en el buffer de teclado la memoria 'inicio'.

```
., 1DB9 LDA #$20
., 1DBB STA $0277,Y
., 1DBE STY $C6
., 1DC0 LDX #$04
., 1DC2 CLC
., 1DC3 LDA $1DDC,X
., 1DC6 ADC $1DE2,X
., 1DC9 CMP #$0A
., 1DCB BCC $1DD3
., 1DCD SEC
., 1DCE SBC #$0A
., 1DD0 SEC
., 1DD1 BCS $1DD4
., 1DD3 CLC
., 1DD4 STA $1DDC,X
., 1DD7 DEX
., 1DD8 BPL $1DC3
., 1DDA RTS
.
.
```

Incrementa la memoria 'inicio' el valor de la memoria 'incremento'.

```
.:1DDB 00 00 00 01 00   → Memoria 'inicio'
.:1DE0 00 01 00 00 00   → Memoria 'incremento'
.:1DE5 01 00 0D 0D 50
.:1DEA 52 49 4D 2E 4C
.:1DEF 49 4E 45 41 2C
.:1DF4 49 4E 43 02 45
.:1DF9 4D 45 4E 54 4F
.:1DEF 3F 0D 2E 2C 20
.
```

De $1DE7 a $1DFF. Esta el texto inicial.



Nombrando programas

Cuando escribas un programa largo y quieras guardarlo en disco, utiliza un nombre corto. Es más rápido y siempre puedes renombrarlo cuando lo finalices.

RUN EE.UU.

Títulos de ficheros

Para distinguir cuando hacemos un VERIFY en cinta si lo que el ordenador encuentra es un fichero o un programa, podemos hacer lo siguiente cuando grabemos un programa:

SAVE "NOMBRE"

Cuando sea un fichero:

SAVE "[RVS ON] NOMBRE"

De este modo el nombre de los ficheros aparece en inverso.

J.E.N. RUN EE.UU.

Scroll de estrellas

Un bonito efecto para gráficos o juegos es el que produce el siguiente programa:

```
10 a=rnd(1)*23
20 ?spc(a)"*"
30 fort=1to40:next
40 goto10
```

El bucle de espera de la línea 30 puede cambiarse para producir otros efectos.

En el caso de que el ordenador que tengas sea un 64, cambia el 23 de la línea 10 por un 40.

G.M. RUN EE.UU.



Artista veloz

Por Bruce Jaeger
(RUN EE.UU)

C-64

Cambiar el color de la pantalla, marco y caracteres en el C-64 no tiene por qué ser pesado. Este programa te permite cambiar los colores en un abrir y cerrar de ojos valiéndote sólo de las teclas Commodore y de función.

Una de las posibilidades más interesantes del Commodore 64 es la variedad de colores que ofrece. Sin embargo, el intérprete Basic del ordenador no facilita precisamente el uso de estos 16 colores.

Al empezar, las letras aparecen en azul claro sobre fondo azul oscuro, lo cual es una combinación difícilmente legible incluso en un monitor decente, por no hablar de un televisor normal. Para cambiar los colores del borde y del fondo, habría que escribir (con POKE) en el registro 53280 un número de 0 a 15 para el marco, y en el 53281 para la pantalla.

El programa en lenguaje máquina que presento a continuación simplifica este proceso. Carga y ejecuta el programa "Cambiador de color" y podrás cambiar el marco, la pantalla, y los colores de los caracteres pulsando una tecla, incluso durante la ejecución de un programa o mientras estás programando.

Para usar el programa, teclea cuidadosamente el programa siguiendo el listado adjunto, y almacénalo en la memoria antes de ejecutarlo. (Es conveniente guardar **todos** los programas que teclees antes de ejecutarlos, especialmente los programas en lenguaje máquina. Un pequeño error de mecanografía podría "bloquear" el ordenador, con lo que tendrías que teclear nuevamente el programa completo).

Ahora ejecuta el programa. Tras un segundo aproximadamente la indicación Ready aparecerá en la pantalla. El programa ha sido cargado en la memoria, y empieza en $C350 (500 decimal) y ha redirigido los vectores de interrupción, para que apunten hacia él.

Ahora, mantén la tecla Commodore pulsada (en el extremo izquierdo de la hilera inferior). Pulsa la tecla de función F1 para cambiar el color del marco. Pulsa la tecla de función F3, y cambia el fondo de la pantalla. La tecla de función F5 cambia el color de todos los caracteres de la pantalla. Y la tecla de función F7 desactiva el programa. (Para volver a activarlo teclea SYS 50000 y pulsa la tecla return).

Elegí la combinación de la tecla, Commodore y las de función porque la probabilidad de que usarlas juntas en un programa es relativamente escasa. De ahora en adelante te resultará fácil cambiar estos colores.

Cómo funciona el programa. "El cambiador de color" es un programa controlado por interrupciones. Sesenta veces por segundo el intérprete Basic del Commodore detiene momentáneamente cualquier tarea que esté realizando en ese momento para llevar a cabo una serie de tareas internas como por ejemplo, actualizar el reloj interno, comprobar si se ha pulsado la tecla stop, etc.

En las posiciones de memoria 788 y 789 ($0314 y $0315 en hexadecimal) hay dos punteros que le indican al ordenador la dirección a la que debe ir para llevar a cabo estas tareas internas. Es posible cambiar estos punteros para que apunten a un programa escrito por ti mismo.

Estudiemos la primera parte del listado de ensamblador, titulado "Cambiar vectores de interrupción". Se trata de un pequeño programa en sí mismo que desactiva todas las interrupciones para que el ordenador no se confunda mientras tú cambias los punteros. Luego cambia dichos punteros para que señalen el primer paso de su programa principal, en este caso $C35D.

Observa que la dirección $C35D se almacena en la memoria "al revés", en lo que se denomina "formato de byte-bajo, byte-alto". $5D se guarda en $0314, y $C3 en $0315. Luego, mediante la instrucción CLI se vuelven a activar las interrupciones y el RTS (retorno de subrutina) devuelve el control al Basic. Ahora, cada 1/60º de segundo, el Basic salta a su propio programa, que empieza en $C35D.

Programa principal. En primer lugar se carga el acumulador con el contenido de SKEY, que contiene el valor de cualquier tecla de cambio, control o Commodore pulsadas. Al pulsar la tecla Commodore carga un 2 en esta dirección. El ordenador comprueba que, realmente, hay un 2 en ese lugar. De lo contrario, el usuario no está pulsando la tecla Commodore, y el programa sabe que no se le necesita, por lo que cede el control a la subrutina Back con BNE BACK (Branch if Not Equal - Saltar si no es igual), que envía al ordenador al resto de las tareas internas.

Cuando **el usuario pulsa** la tecla Commodore, el acumulador se carga con el contenido de Key (valor de toda tecla, distinta de la de control, que se está pulsando en ese momento). El programa compara este valor con FLAG, que contiene el valor de la última tecla analizada por el programa. Si es el mismo, el programa vuelve a la rutina Back, para evitar que el programa realice 60 cambios de color por segundo (¡puede llegar a marear!). Entre uno y otro cambio de color tiene que soltar las teclas.

Si el acumulador no coincide con el anterior valor de Flag lo primero que se hace es guardar el nuevo valor de la tecla en FLAG. Para ello se usa la instrucción STAT FLAG. (Con ello evitará repeticiones de la nueva tecla la siguiente vez). Luego el valor de la tecla se compara con 4, el valor que obtenemos en la posición 197 al pulsar la tecla F1. Si dicho valor es 4, el programa cede el control a BORCOL, la rutina de cambio de color del marco.

Si la tecla no equivale a 4, se compara con 5 (la que tiene el valor de F3), 6 (la tecla F5) y 3 (la tecla F7). Si después de todo esto no obtiene equivalencia, el programa cede el control a la rutina Back.

Subrutinas de color. Tanto BORCOL (cambio de color del borde) como SCNCOL (cambio de color de la pantalla) son muy sencillas. Basta con añadir un 1 a la dirección de memoria correspondiente al color de fondo de la pantalla (53281) o al color del marco (53280), y luego saltar a la rutina Back.

CHARCO (color de los caracteres) cambia el color de los caracteres de la pantalla y es más complicada; es necesario cambiarlos uno por uno.

CURCOL (posición 646 —decimal— de la memoria) contiene el color actual del cursor. En primer lugar se incrementa el CURCOL para obtener el nuevo color y se carga el acumulador con este número (LDA CURCOL).

La sección siguiente de la rutina utiliza el direccionamiento indexado indirecto para recorrer toda la RAM de color del Commodore 64 ($D800 a $DBE7). La dirección inicial de la RAM de color se almacena en el "formato byte-bajo, byte-alto" en las direcciones $FC y $FD de la página cero. (Ver en cualquier manual de programación en lenguaje máquina del 6502 la explicación del direccionamiento indexado indirecto).

El programa luego hace un bucle recorriendo las cuatro páginas de 256 bytes de la RAM de color, almacenando el nuevo color en cada posición. El registro X conserva la página en la que está trabajando la rutina; al alcanzar $DC, se da por terminado el recorrido de la memoria y se cede el control a la rutina Back.

Cómo desactivar el "Cambiador de Color". Para poder escribir o leer otro programa o datos en la zona ocupada por el programa, es necesario desactivarlo. De lo contrario, los vectores de interrupción harán que el ordenador vaya sumiso a esta dirección, y si está introduciendo algo ahí, ¡ZAS! se queda bloqueado instantáneamente. Además, conviene desactivar el programa antes de usar la unidad de cassette. (Para volver a activarlo, teclea SYS 50000). Esta rutina tiene el efecto contrario a la rutina original de preparación: se desactivan las interrupciones y los punteros de interrupción se cargan con su contenido normal, apuntando a la dirección $EA31

Como medida de precaución, siempre que teclea un programa adicional almacénalo antes de ejecutarlo.

```
10 REM *********************
15 REM *                   *
20 REM * CAMBIADOR DE COLOR *
25 REM * ------------------ *
30 REM *    BRUCE JAEGER   *
35 REM *                   *
40 REM *********************
50 REM
55 REM
60 REM RUTINA PARA PONERLO EN MEMORI
A
65 REM
100 DATA120,169,93,141,20,3,169,195,
141,21,3,88,96,173,141,2,201,2,208,6
8
110 DATA165,197,205,186,195,240,61,1
41,186,195,201,4,240,15,201,5,240,17

120 DATA201,6,240,19,201,3,240,45,76
,168,195,238,32,208,76,168,195,238,3
3
130 DATA208,76,168,195,238,134,2,173
,134,2,160,0,132,252,162,216,134,253

140 DATA145,252,200,208,251,232,224,
220,208,244,76,168,195,76,49,234,120

150 DATA169,49,141,20,3,169,234,141,
21,3,88,76,168,195
160 FORX=0TO105:READC:POKE50000+X,C:
SU=SU+C:NEXT
170 IFSU<>14548THENPRINT"ERROR[SPC]E
N[SPC]DATAS":END
180 SYS50000
```

```
LINE# LOC   CODE        LINE

00001  0000             ; ***********************************
00002  0000             ; CODIGO FUENTE DEL CAMBIADOR DE COLOR
00003  0000             ; ***********************************
00004  0000             ; DEFINE DIRECCIONES
00005  0000             ; ***********************************
00006  0000             SKEY   = 653             ; TECLAS DE CONTROL Y SHIFT
00007  0000             KEY    = 197             ; TECLA PULSADA ACTUALMENTE
00008  0000             CURCOL = 646             ; COLOR ACTUAL DEL CURSOR
00009  0000             COLOR  = 55296           ; RAM DE COLOR
00010  0000             BORDER = 53280           ; COLOR DEL BORDE
00011  0000             BKGRND = 53281           ; COLOR DE FONDO
00012  0000             ; CAMBIA VECTORES DE INTERRUPCION
00013  0000             ; ***********************************
00014  0000             *      = 50000           ; COMIENZA ENSAMBLADO AQUI
00015  C350  78                SEI               ; DESHABILITA INTERRUPCIONES
00016  C351  A9 5D             LDA #<PRGAM       ; REORIENTA LOS VECTORES
00017  C353  8D 14 03          STA $0314         ; DE IRQ PARA APUNTAR AL
00018  C356  A9 C3             LDA #>PRGAM       ; PROGRAMA
00019  C358  8D 15 03          STA $0315
00020  C35B  58                CLI               ; HABILITA INTERRUPCIONES
00021  C35C  60                RTS               ; VUELVE A BASIC
00022  C35D             ; PROGRAMA CAMBIADOR DE COLOR
00023  C35D             ; ***********************************
00024  C35D  AD 8D 02   PRGAM  LDA SKEY          ; COMPRUEBA TECLAS SHIFT Y CBM
00025  C360  C9 02             CMP #2            ; TECLA CBM PULSADA ?
00026  C362  D0 44             BNE BACK          ; NO----ENTONCES SALIR
00027  C364  A5 C5             LDA KEY           ; COMPROBAR TECLA PULSADA
00028  C366  CD BA C3          CMP FLAG          ; IGUAL QUE LA ULTIMA ?
00029  C369  F0 3D             BEQ BACK          ; SI---SALIR (EVITAR REBOTES)
00030  C36B  8D BA C3          STA FLAG          ; GUARDARLA PARA LA PROXIMA VEZ
00031  C36E  C9 04             CMP #4            ; TECLA F1?
00032  C370  F0 0F             BEQ BORCOL        ; SI--IR A RUTINA DEL BORDE
00033  C372  C9 05             CMP #5            ; TECLA F3?
00034  C374  F0 11             BEQ SCNCOL        ; SI--IR A RUTINA DE LA PANTALLA
00035  C376  C9 06             CMP #6            ; TECLA F5?
00036  C378  F0 13             BEQ CHARCO        ; SI--IR A RUTINA DE CARACTERES
00037  C37A  C9 03             CMP #3            ; TECLA F7?
00038  C37C  F0 2D             BEQ RESET         ; SI--IR A RUTINA DE RESET
00039  C37E  4C A8 C3          JMP BACK
```

```
C381                ; RUTINAS DE COLORES
C381                ; ************************************
C381                BORCOL          ; RUTINA PARA EL BORDE
C381                ; ************************************
C381  EE 20 D0           INC BORDER      ; CAMBIA EL BORDE
C384  4C A8 C3           JMP BACK        ; SALE
C387                SCNCOL          ; RUTINA PARA EL FONDO
C387                ; ************************************
C387  EE 21 D0           INC BKGRND      ; CAMBIA EL FONDO
C38A  4C A8 C3           JMP BACK        ; SALE
C38D                CHARCO          ; RUTINA PARA LOS CARACTERES
C38D                ; ************************************
C38D  EE 86 02           INC CURCOL      ; CAMBIA EL COLOR DEL CURSOR
C390  AD 86 02           LDA CURCOL      ; LEE EL NUEVO COLOR
C393  A0 00              LDY #00         ; INDICE Y=0
C395  84 FC              STY $FC         ;
C397  A2 D8              LDX #$D8        ; ULTIMO BYTE DE LA RAM DE COLOR
C399  86 FD         LOOP STX $FD         ; LO GUARDA EN $FD
C39B  91 FC         INLOOP STA ($FC),Y   ; PONE EL NUEVO COLOR
C39D  C8                 INY             ; INCREMENTA INDICE Y
C39E  D0 FB              BNE INLOOP      ; SI Y<>0 (O 256) REPITE
C3A0  E8                 INX             ; INCREMENTA X (BYTE ALTO)
C3A1  E0 DC              CPX #$DC        ; SI X=$DC ES QUE HEMOS
C3A3                ; PASADO LA RAM DE COLOR
C3A3  D0 F4              BNE LOOP        ; NO...REPETIR OTRAS 256 VECES
C3A5  4C A8 C3           JMP BACK        ; SI...LO HICIMOS...SALIR
C3A8                ; CONTINUAR LA RUTINA NORMAL DE INTERRUPCION
C3A8                ; ************************************
C3A8  4C 31 EA      BACK JMP $EA31
C3AB                ; QUITAR EL CAMBIADOR DE COLOR
C3AB                ; ************************************
C3AB  78            RESET SEI            ; DESHABILITAR INTERRUPCIONES
C3AC  A9 31              LDA #$31        ; VOLOVER A COLOCAR LOS
C3AE  8D 14 03           STA $0314       ; VECTORES ORIGINALES
C3B1  A9 EA              LDA #$EA
C3B3  8D 15 03         , STA $0315
C3B6  58                 CLI             ; HABILITAR INTERRUPCIONES
C3B7  4C A8 C3           JMP BACK
C3BA                ; ************************************
C3BA                     FLAG *=*+1      ; ALMACENA AQUI EL INDICADOR
C3BB                ; DE REPETICION DE TECLA
C3BB                     .END

ERRORS = 00000

SYMBOL TABLE

SYMBOL VALUE
 BACK     C3A8   BKGRND   D021   BORCOL   C381   BORDER   D020
 CHARCO   C38D   COLOR    D800   CURCOL   0286   FLAG     C3BA
 INLOOP   C39B   KEY      00C5   LOOP     C399   PRGAM    C35D
 RESET    C3AB   SCNCOL   C387   SKEY     028D

END OF ASSEMBLY
```



Magia

Números aleatorios

Escribe el siguiente programa en tu ordenador:

10 fora=1to6:b=int(rnd(1)*100):Print:next

Cada vez que haces RUN, escribes seis números al azar, entre 0 y 99. Ahora apaga el ordenador y repite la operación.

¡El ordenador ha repetido los números! Para evitar que esto ocurra cada vez que haces run en un programa normal, cambia el RND(1) por RND(0).

I.L. RUN EE.UU.

Mnemotécnico

Para conectar el monitor de lenguaje máquina en el VIC-20, hay que hacer SYS32768, pero es más fácil acordarse de SYS 6*4096.

A.I.B. RUN EE.UU.

Caza de errores

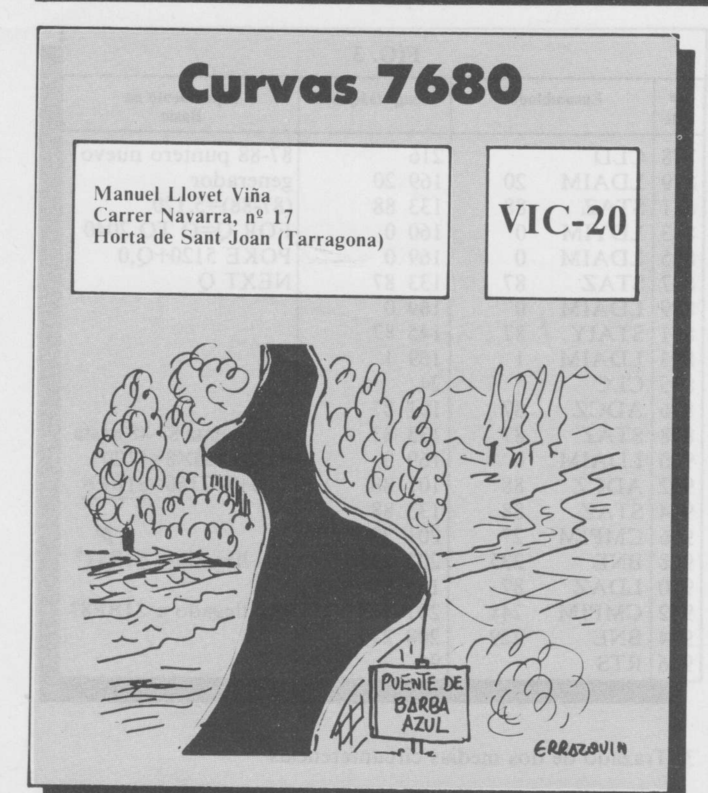
Cuando busques la causa de un syntax error, por ejemplo, pulsa shift y commodore a la vez. De este modo pasas al modo minúsculas y es más fácil distinguir las letras de los números, el "0" de la "o" etc.

M.M. RUN EE.UU.

Sonidos en el VIC

10 ?chr$(204.5+rnd(1));:x=int(rnd(1)*120):Poke36875,15:Poke36875,x:goto10

C.B. RUN EE.UU.



En esta primera colaboración es un programa que creo muy interesante para aquellos que no teniendo el cartucho para dibujar en alta resolución quieran usar el VIC-20 para hacer aparecer en la pantalla una curva introduciendo su ecuación matemática. Para los que no sean muy amigos de las matemáticas pueden descubrir el secreto de la alta resolución y usarlo en otros menesteres. Ante todo indicar que este programa sólo sirve para el VIC-20 con memoria MINIMA, sin ningún tipo de ampliación. El programa se ve completado con unas subrutinas en lenguaje de máquina para hacer que su velocidad sea elevada y el usuario no se dé cuenta de cuándo se ejecutan, pues su misión es poco vistosa. Dado que el problema principal es la falta de memoria, dicho programa en lenguaje de máquina se introduce en memoria leyendo un fichero, pues de hacerse con DATA's el programa no entraría.

El programa BASIC de la figura 1 comienza leyendo dicho fichero y colocando los datos en un principio en memoria a partir de la 5.120. A continuación, líneas 12 a 16, estos datos son trasladados al buffer del cassette a partir de la 828. Esto debe hacerse así puesto que en la primera lectura el buffer del cassette está siendo usado para la propia lectura de datos.

La forma de dibujar en pantalla en alta resolución consiste en usar un generador de caracteres propio. Empezamos llenando la pantalla con los 255 primeros caracteres, delimitamos la pantalla a 11 líneas, cambia la dirección del generador de caracteres a la posición 5.120 y baja el borde de la pantalla hasta centrarla.

Para 11 líneas de pantalla necesitamos un generador de caracteres de 2K RAM, desde 5120 hasta 7160, viniendo la memoria del VIC justísima. Si tenemos los dos K RAM del generador de caracteres a cero, aunque la pantalla esté llena de caracteres, al ser todos blancos o sea un espacio, la pantalla parecerá vacía. Si ahora llenamos con ciertos códigos nuestro nuevo generador de caracteres al estar la pantalla cubierta de letras, irán apareciendo cosas en pantalla. Todo consiste pues en definir correctamente el generador de caracteres de forma que su representación en pantalla coincida con la curva que deseamos diseñar. La subrutina en lenguaje de máquina que comienza en 878 limpia el generador de caracteres poniendo la pantalla en blanco.

Las líneas 18 y 20 indican al usuario que la ecuación de la curva Y=F(X) se debe de escribir en alguna de las líneas 43,44 o 45 cuando este programa se detiene en el LIST. Una vez escrita la ecuación se debe de arrancar el programa con GOTO 22. Si en este momento se hiciera RUN el programa leería de nuevo el fichero machacando nuestro lenguaje de máquina situado en el buffer.

Las líneas 22 y 30 preguntan sobre los valores máximos y mínimos de la variable X y la función Y. Esto se necesita para encuadrar la función en la pantalla. Si respondemos "S" a SUPERPOS, entonces la curva se dibujará superpuesta a la que se dibujó anteriormente.

Además del VIC realiza un cambio de variable para adaptar cualquier curva a su pequeña pantalla de 11 líneas con sus dimensiones de 89 pixels de alta por 176 de larga.

Al finalizar el trazado de la curva el VIC espera que se pulse cualquier tecla, volviendo al modo texto.

Si alguna vez al introducir una curva el VIC escribe algo ininteligible y se queda en el modo de alta resolución es que ha querido poner datos fuera del generador de caracteres y la línea 50 lo ha parado con un STOP. En este caso pulsar a la vez RUN/STOP y RESTORE y arrancar de nuevo el programa con GOTO 22.

Es conveniente que copies el fichero justo detrás del programa BASIC para que cuando este último se ejecute y os pida la lectura, el fichero esté a continuación ahorrándonos un cambio de cinta.

Para copiar el fichero en cinta usad el siguiente programa:

```
10 OPEN 1,1,1,"LENMAQ"
20 FOR I=O TO 88
30 INPUT"DATO";A
40 PRINT#1,A
50 NEXT
70 CLOSE 1
80 STOP
```

Para que vayais practicando os mando una serie de ecuaciones matemáticas con los respectivos valores máximos y mínimos de X e Y. Espero que os guste mi plotter-programa.

FIG. 1

```
1 PRINT"[CLR][3CRSRD][3SPC]ESPERE[SPC]
UN[SPC]MOMENTO"
2 OPEN1,1,0,"LENMAQ"
4 FORQ=0TO88
6 INPUT#1,B
8 POKE5120+Q,B
10 NEXT:CLOSE1
12 FORQ=0TO88
14 POKE828+Q,PEEK(5120+Q)
16 NEXT:SYS(878)
18 PRINT"[CLR][2CRSRD]**CURVAS**"
20 PRINT"[2CRSRD][2SPC]43-45Y=F(X)[2SPC]
GOTO22":LIST43-45
22 INPUT"[CRSRD]YMAX[SPC]YMIN";Q,R
24 F(8)=0:F(0)=128:F(1)=64:F(2)=32:F
(3)=16
26 F(4)=8:F(5)=4:F(6)=2:F(7)=1
28 R=11/(Q-R)
30 INPUT"[CRSRD]XMAX[SPC]XMIN";S,T
32 S=(S-T)/175
34 INPUT"[CRSRD]SUPERPOSI";A$
36 IFA$="N"THENSYS(878)
38 SYS(828)
40 FORC=0TO175
42 X=T+S*C
44 Y=SIN(X)
46 Y=R*(Q-Y):B=INT(Y):Y=(Y-B)*8
48 D=C/8:E=INT(D):B=5120+176*B+E*8+Y

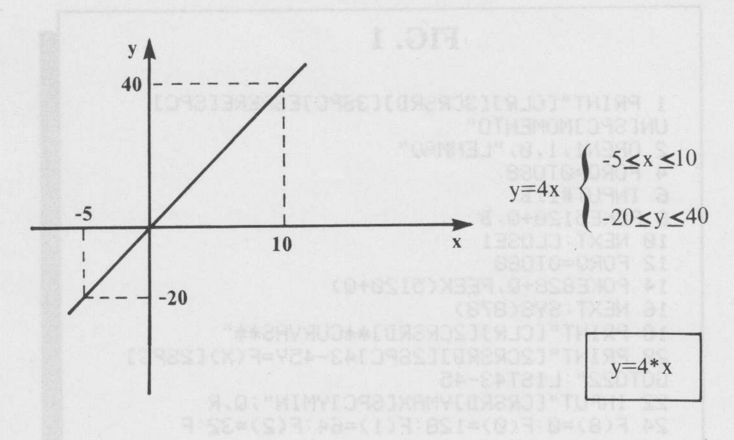
50 D=INT((D-E)*8):IFB<5100ORB>7200TH
ENPRINT"ERROR":STOP
52 POKEB,PEEK(B)ORF(D):NEXT
54 GETA$:IFA$=""THEN54
56 Q=36865:POKEQ,38:POKEQ+1,150
58 POKEQ+2,174:POKEQ+4,240:GOTO18
```

FIG. 2

Nº byte	Ensamblador		Leng. máquina	Equivalente en Basic
828	LDAIM	60	169 60	POKE36865,60
830	STA	36865	141 1 144	borde superior pantalla
833	LDYIM	0	160 0	89-90 puntero RAM
835	STYZ	87	132 87	COLOR
837	STYZ	89	132 89	(89-90)=38400=$9600
839	LDAIM	30	169 30	(87-88)=7680=$1E00
841	STAZ	88	133 88	87-88 puntero RAM
843	LDAIM	150	169 150	SCREEN
845	STAZ	90	133 90	
847	LDXIM	0	162 0	FOR Q=0 TO 255
849	TYA		152	POKE 7680+QQ
850	STAIY	87	145 87	POKE38400+Q,2
852	LDAIM	2	169 2	NEXT Q
854	STAIY	89	145 89	llena las 11 líneas de
856	INY		200	255 caracteres
857	BNE	246	208 246	
859	LDAIM	253	169 253	POKE 36869,253
861	STA	36869	141 5 144	dirección del generador de caracteres en 5120
864	LDAIM	128	169 128	POKE36866,PEEK
866	ORA	36866	13 2 144	(36866) OR128
869	STA	36866	141 2 144	matriz de video 7680
872	LDAIM	150	169 150	POKE 36867,150
874	STA	36867	141 3 144	11 líneas de pantalla
877	RTS		96	

Ejemplos de curvas interesantes

1. Trazado de una recta que pasa por el origen



2. Trazado del eje X

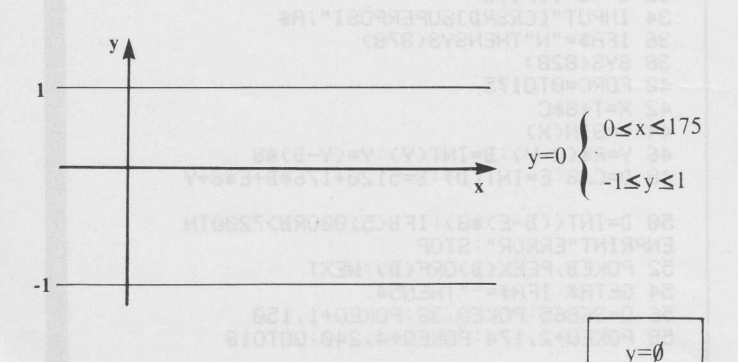
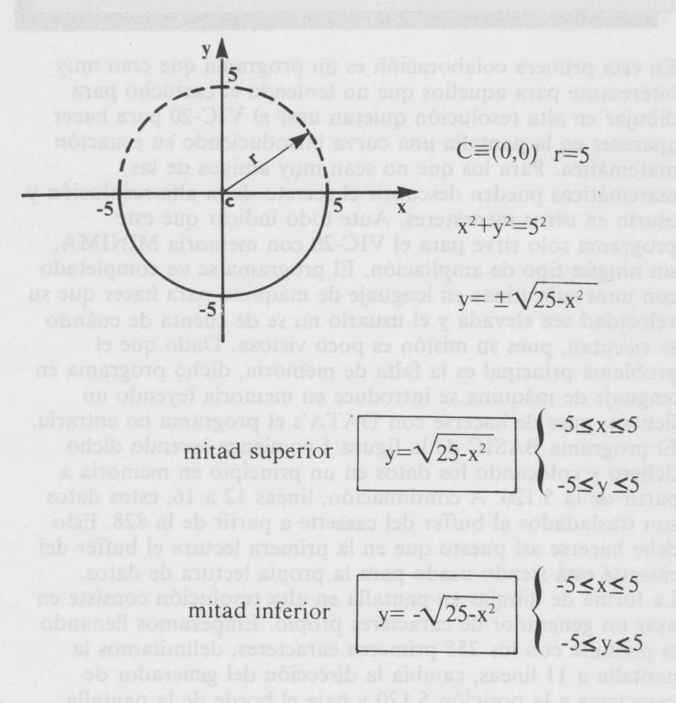


FIG. 3

Nº byte	Ensamblador		Leng. máquina	Equivalente en Basic
878	CLD		216	87-88 puntero nuevo
879	LDAIM	20	169 20	generador
881	STAZ	88	133 88	(87-88)=5.120
883	LDYIM	0	160 0	FOR Q=O TO 2040
885	LDAIM	0	169 0	POKE 5120+Q,0
887	STAZ	87	133 87	NEXT Q
889	LDAIM	0	169 0	
891	STAIY	87	145 87	
893	LDAIM	1	169 1	
895	CLC		24	
896	ADCZ	87	101 87	
898	STAZ	87	133 87	incrementa 87-88 hasta
900	LDAIM	0	169 0	5120+255×8=5120
902	ADCZ	88	101 88	+2040=7160=$1BF8
904	STAZ	88	133 88	
906	CMPIM	27	201 27	
908	BNE	235	208 235	¿ha llegado a $1Bxx?
910	LDAZ	87	165 87	
912	CMPIM	248	201 248	¿ha llegado a $1BF8?
914	BNE	229	208 229	
916	RTS		96	

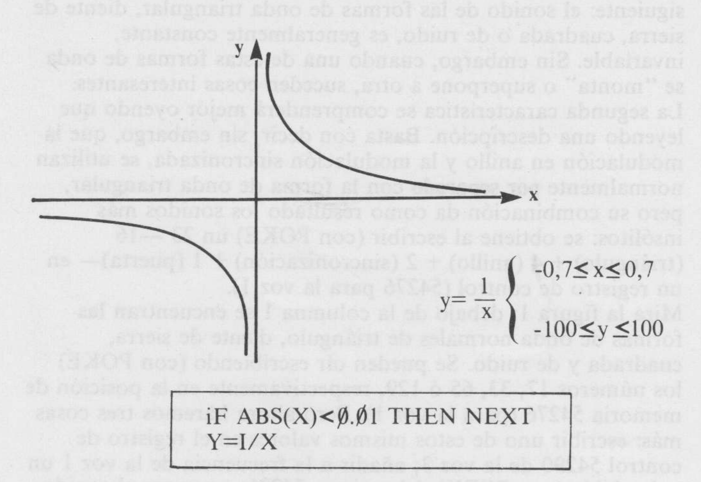
3. Trazado de dos medias circunferencias



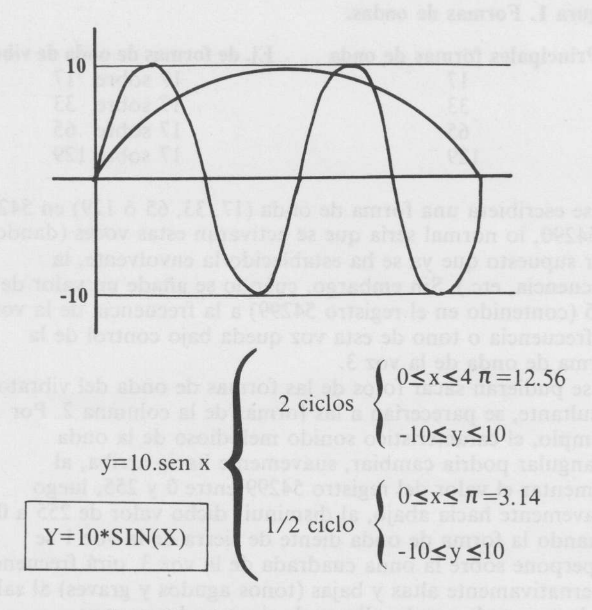
Se observará una excentricidad exagerada que la circunferencia no debería de presentar. La deformación se debe al hecho de que la forma de la pantalla es rectangular y no cuadrada, con lo que las dimensiones en x e y, no son iguales.

Para obtener la circunferencia entera, se pondrá la primera ecuación, la correspondiente al signo más, se introducirán los límites y una vez representada se meterá la del signo menos, con los mismos límites que en el caso anterior y a la pregunta de SUPERPOS se contestará afirmativamente con una S.

4. Trazado de una hipérbola equilátera.

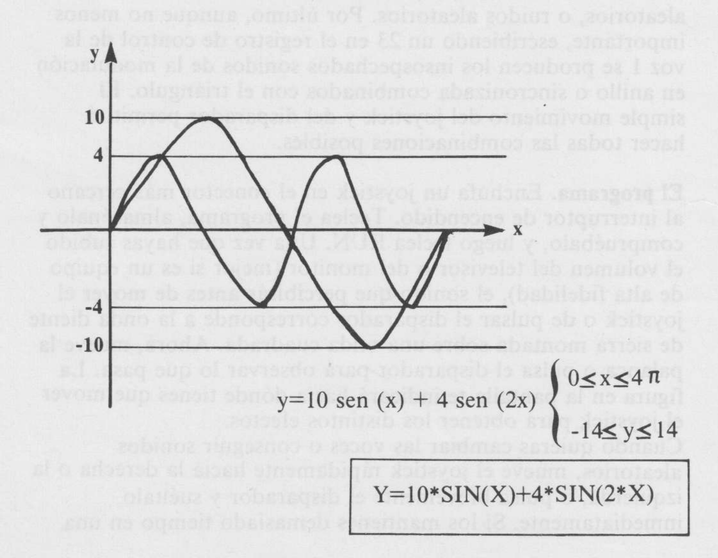


5. Trazado de una sinusoide (2 ciclos y medio ciclo).



En este caso se ha utilizado una escala diferente en cada caso para ver de la misma curva dos partes distintas. Es como si en el segundo caso se viese con "lupa" el primer semiciclo, para apreciar el detalle.

6. Trazado de la suma de dos sinusoides de frecuencia distinta.



Resulta interesante representar las tres sinusoides superpuestas la una con la otra: escribir primero 10sen(x), luego 4sen(2x) y por último la suma de ambas; en los tres casos con las mismas limitaciones de x e y.

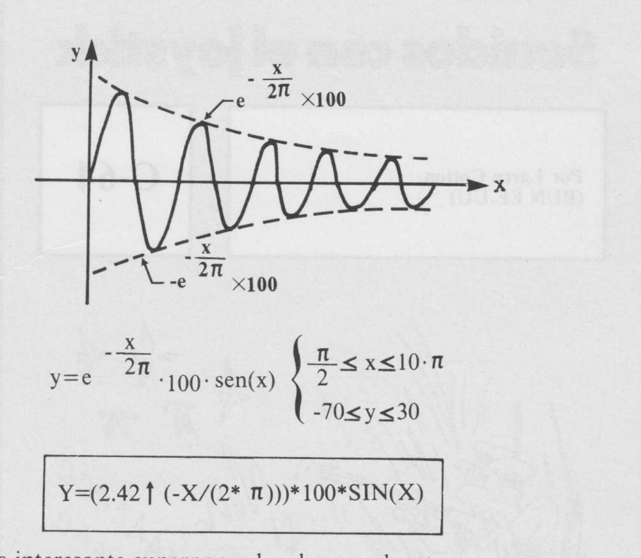
7. Trazado del producto de dos senos de frecuencia distintas.

y=10 sen(x).sen (4x) $\begin{cases} 0 \leq x \leq 4\pi = 12{,}56 \\ -10 \leq y \leq 10 \end{cases}$

```
Y=10*SIN(X)*SIN(4*X)
```

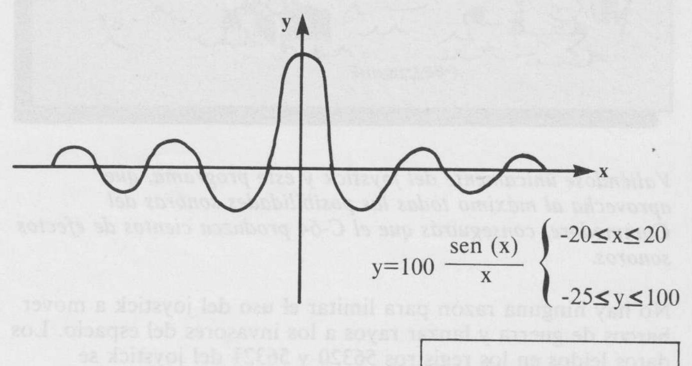
Como en el caso anterior resulta interesante superponer las tres sinusoides con los mismos limites para las tres.

8. Trazado de una sinusoide amortiguada.

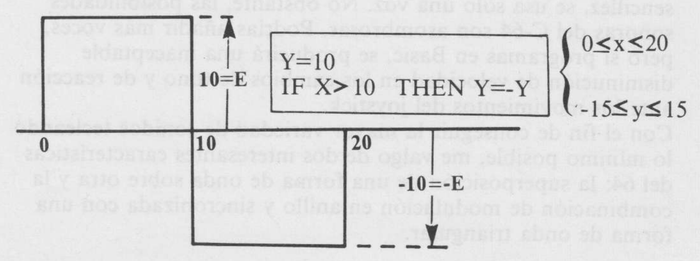


Es interesante superponer las dos envolventes.

9. Trazado de (sen(x))/x



10. Trazado de una señal cuadrada



11. Trazado de una señal cuadrada por su desarrollo de Fourier

$$\text{señal cuadrada}=\sum_{n=1}^{\infty} \frac{4E}{n\pi}\ \text{sen}\,(n\,x)$$ n sólo toma valores impares 1, 3, 5, 7,...

si por comodidad tomamos la amplitud $E=\frac{100\pi}{4}$ queda

$$\sum_{n=1}^{N} \frac{100}{n}\ \text{sen}\,(n.x) \rightarrow$$

```
Y=0:FOR N=1 TO NØ STEP2
Y=Y+100*SIN(N*X)/N:NEXTN
```

$0 \le \quad \le 2\pi$ $-110 \le y \le 110$

En la teoría la sumatoria se extiende hasta infinito, cosa que en la práctica no es posible. Se escoge un número máximo de armónicos NØ. Resulta interesante ver cómo la señal se aproxima más y más a una señal cuadrada dando a NØ valores de 2, 3, 5, 10, 30, etc. Observar los frentes de subida y bajada, y las sinusoides amortiguadas de las partes superior e inferior.



Valiéndose únicamente del joystick y este programa, que aprovecha al máximo todas las posibilidades sonoras del Commodore, conseguirás que el C-64 produzca cientos de efectos sonoros.

No hay ninguna razón para limitar el uso del joystick a mover barcos de guerra y lanzar rayos a los invasores del espacio. Los datos leídos en los registros 56320 y 56321 del joystick se pueden usar para controlar imágenes, tanto visuales como sonoras. Este programa muestra cómo se pueden controlar los sonidos utilizando sólo el joystick.
Para que el programa no resulte muy largo y no pierda sencillez, se usa sólo una voz. No obstante, las posibilidades sonoras del C-64 son asombrosas. Podrías añadir más voces, pero si programas en Basic, se producirá una inaceptable disminución de velocidad en los cambios de tono y de reacción ante los movimientos del joystick.
Con el fin de conseguir la mayor variedad de sonidos tecleando lo mínimo posible, me valgo de dos interesantes características del 64: la superposición de una forma de onda sobre otra y la combinación de modulación en anillo y sincronizada con una forma de onda triangular.

Para que suene bien. La primera característica significa lo siguiente: el sonido de las formas de onda triangular, diente de sierra, cuadrada o de ruido, es generalmente constante, invariable. Sin embargo, cuando una de estas formas de onda se "monta" o superpone a otra, suceden cosas interesantes.
La segunda característica se comprenderá mejor oyendo que leyendo una descripción. Basta con decir, sin embargo, que la modulación en anillo y la modulación sincronizada, se utilizan normalmente **por separado** con la forma de onda triangular, pero su combinación da como resultado los sonidos más insólitos: se obtiene al escribir (con POKE) un 23 —16 (triángulo) + 4 (anillo) + 2 (sincronización) + 1 (puerta)— en un registro de control (54276 para la voz 1).
Mira la figura 1: debajo de la columna 1 se encuentran las formas de onda normales de triángulo, diente de sierra, cuadrada y de ruido. Se pueden oír escribiendo (con POKE) los números 17, 33, 65 ó 129, respectivamente en la posición de memoria 54276 (para la voz 1). Pero ahora haremos tres cosas más: escribir uno de estos mismos valores en el registro de control 54290 de la voz 3; añadir a la frecuencia de la voz 1 un valor leído (con PEEK) del registro 54296; y apagar el sonido de la voz 3 escribiendo 143 en 54296.

Figura 1. Formas de ondas.

Principales formas de onda	Ej. de formas de onda de vibrato
17	17 sobre 17
33	17 sobre 33
65	17 sobre 65
129	17 sobre 129

Si se escribiera una forma de onda (17, 33, 65 ó 129) en 54276 y 54290, lo normal sería que se activaran estas voces (dando por supuesto que ya se ha establecido la envolvente, la frecuencia, etc.). Sin embargo, cuando se añade un valor de 0 a 255 (contenido en el registro 54299) a la frecuencia de la voz 1, la frecuencia o tono de esta voz queda bajo control de la forma de onda de la voz 3.
Si se pudieran sacar fotos de las formas de onda del vibrato resultante, se parecerían a las formas de la columna 2. Por ejemplo, el característico sonido melodioso de la onda triangular podría cambiar, suavemente hacia arriba, al aumentar el valor del registro 54299 entre 0 y 255, luego suavemente hacia abajo, al disminuir dicho valor de 255 a 0. Cuando la forma de onda diente de sierra de la voz 1 se superpone sobre la onda cuadrada de la voz 3, oirá frecuencias alternativamente altas y bajas (tonos agudos y graves) al salirse de la meseta las ondas diente de sierra, y luego caer abruptamente al llano para sonar ahí. El valor en 54299 pasa de 255 a 0, y luego vuelve a 255 directamente. Este es el primer sonido que se oye en el programa.
Cuando la principal forma de onda es la de ruido (129), se oirán ruidos, obviamente, pero cuando la forma de onda que controla el vibrato (voz 3) es ruido, se oirá una serie de tonos aleatorios, o ruidos aleatorios. Por último, aunque no menos importante, escribiendo un 23 en el registro de control de la voz 1 se producen los insospechados sonidos de la modulación en anillo o sincronizada combinados con el triángulo. El simple movimiento del joystick y del disparador permitirá hacer todas las combinaciones posibles.

El programa. Enchufa un joystick en el conector más cercano al interruptor de encendido. Teclea el programa, almacénalo y compruébalo, y luego teclea RUN. Una vez que hayas subido el volumen del televisor o del monitor (mejor si es un equipo de alta fidelidad), el sonido que percibirás antes de mover el joystick o de pulsar el disparador corresponde a la onda diente de sierra montada sobre una onda cuadrada. Ahora, mueve la palanca o pulsa el disparador para observar lo que pasa. La figura en la pantalla te indicará hacia dónde tienes que mover el joystick para obtener los distintos efectos.
Cuando quieras cambiar las voces o conseguir sonidos aleatorios, mueve el joystick rápidamente hacia la derecha o la izquierda, o pulsa brevemente el disparador y suéltalo inmediatamente. Si los mantienes demasiado tiempo en una

posición, las voces experimentarán algún cambio adicional.
Las líneas 10-90 preparan el formato de la pantalla. Las líneas 100-150 borran y escriben (con POKE) en el chip de sonido los valores adecuados. La línea 115 escribe una frecuencia en la voz 3 y la frecuencia de corte del filtro; las líneas 120 y 130 escriben 8 en los registros de amplitud de impulsos de las voces 1 y 3. De este modo se asegura que la onda cuadrada principal tenga un bello sonido hueco y que la onda cuadrada de la voz 3 haga que el tono de la voz 1 tenga una duración igual hacia arriba y hacia abajo.
Para cambiar la relación de tiempo hacia arriba y hacia abajo (tono alto a bajo), prueba a escribir 1 a 15 en la posición de memoria 54289. Las líneas 140 y 150 fijan una envolvente de volumen constante para la voz 1 y escriben un 143 (15 para el máximo volumen y bit 7 activado: sonido apagado de la voz 3) en 54296.
Las líneas 160-215, asignan y comprueban la validez de los valores de algunas variables que luego se habrán de escribir en el chip de sonido. (Observa que Q y R no se escriben directamente: ver línea 250).
Las líneas 216 y 218 muestran el estado de las variables en la pantalla. Las líneas 220-240 escriben en el chip de sonido las variables cambiantes: dos formas de onda (la principal en 54276 y vibrato en 54290) y la velocidad (la frecuencia de la voz 3 en 54286).
Las líneas 250-280 forman un bucle de 1000 ciclos, cuando no se toca ni el joystick ni el disparador. Para cada ciclo se genera una nueva frecuencia, W, y se escribe en uno de los registros de frecuencia de la voz 1. (El otro, 54272, siempre permanece en cero).
Mientras tanto, el programa analiza (con PEEK) continuamente el registro 56320 del joystick. W se basa en dos valores controlados por el joystick, el tono R (con el joystick en diagonal hacia un lado) y la profundidad Q (con el joystick en diagonal hacia el otro lado), más el valor leído con PEEK en el registro 54299 multiplicado por una constante. Si J permanece en 127 (estando todos los interruptores del joystick abiertos), el programa vuelve a la línea 250 y continúa el interesante efecto de vibrato.
Las líneas 280-380 comprueban el estado del joystick y del disparador. Observa que es muy sencillo detectar cuándo se está pulsando el disparador o el joystick, y en qué dirección. Observa también que en las sentencias REM se describe lo que está pasando.
Las líneas 370-400 proporcionan el sonido aleatorio que el ordenador va a producir cuando pulsas la palanca hacia la izquierda.
Las líneas 500-530 y 600-640 son subrutinas que recorren las formas de onda cuando se pulsa el disparador o cuando se mueve la palanca hacia la derecha. Las líneas 1000-1015 y 2000-2020 son subrutinas cuya ejecución se desencadena aleatoriamente por las sentencias GOSUB de las líneas 390 y 400.
Presta atención siempre a los valores que cambian en la pantalla según se mueve la palanca o pulsa el disparador. Apunta los sonidos que prefieres para poder repetirlos. Por orden de aparición, los siguientes son ajustes interesantes: 420, 20, 184, 33, 33 ó 600, 80, 150, 23, 17 ó 700, 50, 150, 23, 33 ó 100, 60, 10, 129 y 17. Tómate libertad para experimentar y añade un poco de alegría a tus programas de sonido.

```
10 PRINTCHR$(147)
20 FORT=1TO6:PRINT:NEXT
30 PRINT"[15SPC]ALEATORIO[SPC][SHIFT*]
[SHIFTQ]"
35 PRINT
40 PRINT"[5SPC][SHIFTM][8SPC][SHIFTN]
[5SPC][SHIFTB]"
50 PRINT"[6SPC][SHIFTQ][6SPC][SHIFTQ]
[6SPC][SHIFTQ][5SPC][SHIFTQ][SHIFT*]
[2SPC]FUEGO"
60 PRINT"[7SPC][SHIFTM][4SPC][SHIFTN]
[7SPC][SHIFTB]"
70 PRINT"[32SPC]"
80 PRINT"[4SPC]PITCH[2SPC]DEPTH[2SPC]
SPEED[2SPC]MAIN[2SPC]VIBE"
90 PRINT
100 FORL=54272TO54295:POKEL,0:NEXT
110 P=54299:V=54273
115 POKE54292,50:POKE54287,5
120 POKE54275,8:REM PARA USAR CON/65
V1
130 POKE54289,8:REM PARA USAR CON/65
V3
140 POKE54277,4:POKE54278,89
150 POKE54296,143:REM V3 QUITA VOLUM
EN
160 Q=2:REM DEPTH (PROFUNDIDAD)
170 R=200:REM PITCH (NOTA)
180 S=50:REM SPEED (VELOCIDAD)
190 T=65:REM VIB WAVE (ONDA DE VIBRA
TO)
200 U=33:REM MAIN WAVE (ONDA PRICIPA
L)
210 IFR<0THENR=0
212 IFS<0THENS=0
213 IFS>255THENS=255
214 IFU<1THENU=1
215 IFU>255THENU=255
216 PRINTTAB(4)R"[CRSRL][2SPC]"TAB(1
2)INT(Q*10)"[CRSRL][2SPC]"TAB(19)S"[CRSRL]
[2SPC]";
218 PRINTTAB(25)U"[CRSRL][2SPC]"TAB(
31)T"[CRSRL][2SPC][CRSRU]"
220 POKE54286,S:REM VELOCIDAD DEL VI
BRATO
230 POKE54290,T:REM FORMA DE ONDA DE
L VIBRATO
240 POKE54276,U:REM FORMA DE ONDA PR
INCIPAL
250 FORZ=1TO1000:W=INT((R+PEEK(P)*Q)
/20)
255 IFW>255THENW=255
256 IFW<1THENW=1
260 POKEV,W
270 J=PEEK(56320)
280 IFJ=127THENNEXT
290 IFJ=126THENS=S+2:GOTO210:REM SUB
E
300 IFJ=125THENS=S-2:GOTO210:REM BAJ
A
310 IFJ=111THENGOSUB500:GOTO210

320 IFJ=119THENGOSUB600:GOTO210

330 IFJ=117THENR=R-50:GOTO210

340 IFJ=122THENR=R+50:GOTO210

350 IFJ=118THENQ=Q+.3:GOTO210

360 IFJ=121THENQ=Q-.3:GOTO210

370 IFJ=123THENR=INT(1000*RND(1))+1:
S=INT(250*RND(1))+1:Q=INT(10*RND(1))
+.1
380 IFJ=123THENI=INT(5*RND(1))+1:K=I
NT(4*RND(1))+1
390 ONIGOSUB1000,1005,1010,1015
400 ONKGOSUB2000,2005,2010,2015,2020

410 GOTO210
500 IFT=17THENT=33:RETURN
510 IFT=33THENT=65:RETURN
520 IFT=65THENT=129:RETURN
530 IFT=129THENT=17:RETURN
600 IFU=17THENU=33:RETURN
610 IFU=33THENU=65:RETURN
620 IFU=65THENU=129:RETURN
630 IFU=129THENU=23:RETURN
640 IFU=23THENU=17:RETURN
1000 T=17:RETURN
1005 T=33:RETURN
```

```
1010 T=65:RETURN
1015 T=129:RETURN
2000 U=17:RETURN
2005 U=33:RETURN
2010 U=65:RETURN
2015 U=129:RETURN
2020 U=23:RETURN
```

Memory game

Alvaro Ibáñez
Cardenal Siliceo, 20
28002 Madrid

VIC-20 sin expansión

Este es un juego para ejercitar nuestra memoria visual. En la pantalla aparecen 35 cartas boca abajo. Cada una de ellas contiene unos símbolos, y cada carta que queremos destapar, pulsamos RETURN y hacemos lo mismo con otra carta.
Si las dos cartas son iguales, sumamos un punto y volvemos a intentarlo, en caso contrario será el turno del ordenador.
La mayoría de las veces nos ganará el ordenador, porque almacena en su memoria todas las cartas que van saliendo.
Hay que tener en cuenta una cosa muy importante: El ordenador NO PUEDE hacer trampas, porque no está programado para ello. Lo que ocurre es que, a veces, tiene bastante suerte y se va encontrando parejas.
Hay una carta en blanco que no tiene pareja, por lo que sólo sirve para despistar.
Gana el que mayor número de parejas encuentre.

```
1 REM(C)ALVARO IBANEZ
2 DIMA%(34),A(34),B(34),O(5),S(5)
3 FORI=0TO5:READO(I):NEXT
4 FORI=0TO5:READS(I):NEXT
5 GOTO1000
8 CARDS=35:X=3:Y=2
9 Y$="[HOM][22CRSRD]"
10 C=-1
40 FORI=7725TO8110STEP88:FORJ=ITOI+1
9STEP3
45 C=C+1:A(C)=J
50 FORK=0TO5:POKEJ+O(K),160:NEXT:NEX
T:NEXT
80 FORI=0TO33STEP2:READA:A%(I)=A:A%(
I+1)=A:NEXT:A%(34)=32
85 GOSUB950
90 GOTO2000
100 PLAYER=1
105 M=Y*7+X:CH=PEEK(7725+X*3+Y*88)
110 FORI=0TO5:POKEA(M)+O(I),S(I):NEX
T
150 GETA$:IFA$=""THEN150
160 IFASC(A$)=13ANDCH=160THEN300
180 FORI=0TO5:POKEA(M)+O(I),CH:NEXT
190 X=X+(A$="[CRSRL]")-(A$="[CRSRR]"
)
200 Y=Y+(A$="[CRSRU]")-(A$="[CRSRD]"
)
210 IFX<0THENX=6
220 IFX>6THENX=0
230 IFY<0THENY=4
240 IFY>4THENY=0
260 GOTO100
300 IFW=0THENGOSUB400:W=1:GOTO100
310 GOSUB500
320 IFR=1THENR=0:W=0:GOTO100
330 GOTO700
400 B(M)=A%(M)
410 FORI=0TO5:POKEA(M)+O(I),A%(M):NE
XT
420 FORT=0TO999:NEXT
430 E=A%(M):L=M
440 RETURN
500 B(M)=A%(M)
510 FORI=0TO5:POKEA(M)+O(I),A%(M):NE
XT
520 FORT=0TO999:NEXT
530 IFA%(M)=ETHEN600
540 FORI=0TO5:POKEA(M)+O(I),160:POKE
A(L)+O(I),160:NEXT
550 RETURN
600 FORI=0TO5:POKEA(M)+O(I),32:POKEA
(L)+O(I),32:NEXT
610 IFPLAYER=1THENSC=SC+1
620 IFPLAYER=2THENCC=CC+1
630 CARDS=CARDS-2
650 R=1
660 B(M)=-1:B(L)=-1
662 S$=RIGHT$(STR$(SC),LEN(STR$(SC))
-1):IFLEN(S$)=1THENS$="0"+S$
664 C$=RIGHT$(STR$(CC),LEN(STR$(CC))
-1):IFLEN(C$)=1THENC$="0"+C$
666 PRINTY$;"[2SPC]TU[SPC]";S$;"[3SPC]
";C$;"[SPC]VIC[5SPC][HOM]"
669 IFCARDS=1THEN900
670 RETURN
700 PLAYER=2
705 FORI=0TO34
710 IFB(I)<1THEN760
720 FORJ=0TO34
730 IFJ=ITHEN750
740 IFB(I)=B(J)THEN790
750 NEXTJ
760 NEXTI
770 I=INT(RND(1)*35)
780 IFB(I)THEN770
790 M=I:GOSUB400
800 FORI=0TO34
810 IFI=MTHEN830
820 IFB(I)=B(M)THEN860
830 NEXTI
840 I=INT(RND(1)*35)
850 IFB(I)THEN840
860 M=I:GOSUB500
870 IFR=1THENR=0:GOTO700
880 W=0
890 GOTO100
900 IFSC>CCTHEN930
905 FORI=1TO6
910 PRINTY$;"[10SPC]";C$;"[SPC]VIC[5SPC]
[HOM]"
915 FORT=1TO500:NEXT
920 PRINTY$;"[21SPC][HOM]"
925 FORT=1TO500:NEXT
927 NEXT:GOTO948
```

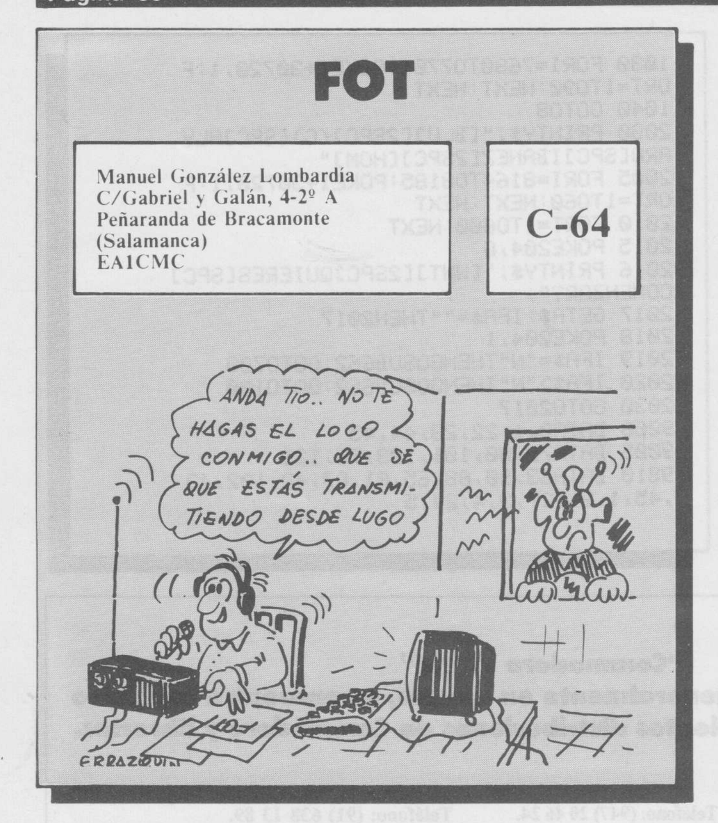
```
930 FORI=1TO6
935 PRINTY$;"[2SPC]TU[SPC]";S$;"[14SPC]
[HOM]"
940 FORT=1TO500:NEXT
945 PRINTY$;"[21SPC][HOM]"
946 FORT=1TO500:NEXT
947 NEXT
948 PRINTY$;"[3SPC]PULSA[SPC]UNA[SPC]
TECLA[3SPC][HOM]":GETA$:IFA$=""THEN9
48
949 RUN
950 FORI=1TO50
960 A=INT(RND(1)*35)
970 B=INT(RND(1)*35)
980 J=A%(A):A%(A)=A%(B):A%(B)=J
985 NEXT
990 RETURN
1000 PRINT"[CLR]"
1010 POKE36879,110
1020 PRINT"[HOM][BLU][RVSON][5SPC]ME
MORY[2SPC]GAME[5SPC][WHT]"
```

```
1030 FORI=7680TO7701:POKEI+30720,1:F
ORT=1TO90:NEXT:NEXT
1040 GOTO8
2000 PRINTY$;"[BLU][2SPC](C)[SPC]ALV
ARO[SPC]IBANEZ[2SPC][HOM]"
2005 FORI=8164TO8185:POKEI+30720,1:F
ORT=1TO60:NEXT:NEXT
2010 FORT=1TO600:NEXT
2015 POKE204,0
2016 PRINTY$;"[WHT][2SPC]QUIERES[SPC]
COMENZAR?";
2017 GETA$:IFA$=""THEN2017
2018 POKE204,1
2019 IFA$="N"THENGOSUB662:GOTO700
2020 IFA$<>"N"THENGOSUB662:GOTO100
2030 GOTO2017
9000 DATA0,1,22,23,44,45
9005 DATA79,80,101,103,76,122
9010 DATA83,90,88,65,81,87,42,102,43
,45,1,2,127,3,4,24,5
```

FOT

Manuel González Lombardía
C/Gabriel y Galán, 4-2º A
Peñaranda de Bracamonte
(Salamanca)
EA1CMC

C-64



Este es un corto programa (1.781 bytes) muy sencillo en el que incluyo algunos de los PRINTS del artículo "PRINTS SOFISTICADOS" y que puede venirle bien a cualquier radioaficionado, sirve para calcular la Frecuencia Optima de Trabajo, a una hora determinada y en una estación del año determinada. También calcula la Frecuencia Máxima Utilizable que está generalmente un 30% por encima. También calcula la Frecuencia Máxima Posible que es alrededor de un 50% superior a la FOT.

Hay que puntualizar que este programa aunque usa unas fórmulas muy sencillas, calcula la FOT con bastante aproximación, ya que todos los cálculos en los que influyen las capas atmosféricas, la radiación solar, etc., son de tipo estadístico y pueden variar...

Cuando el programa pide, mediante INPUT, que se introduzca la hora GMT, DIA, MES, AÑO hay que meter los datos separados por comas; para los no radioaficionados que tengan la curiosidad de ver los PRINTS en acción, les diré que la hora GMT (tiempo del Meridiano de Greenwich) es la hora solar, por lo que a la hora que marque el reloj hay que restarle 2 horas en verano y 1 hora en invierno.

También veréis que cuando pide estos cuatro datos, la hora la pide desde las 00.00 a las 26.00, no es un error, es conveniencia de las fórmulas ¡creéroslo, por favor!

También tener en cuenta que si os da una FOT de 9 MHz (Megahercios) ó 26 MHz, podéis estar seguros que podéis trabajar perfectamente las bandas de 40 (7 MHz) y 10 metros (28 MHz) respectivamente.

Bueno, no me queda más que agradecer a Pere Masats sus artículos y a Francisco J. Dávila (EA8EX) por sus estupendos artículos de programación en una revista especializada de radioafición, sin ellos este programa no habría sido posible.

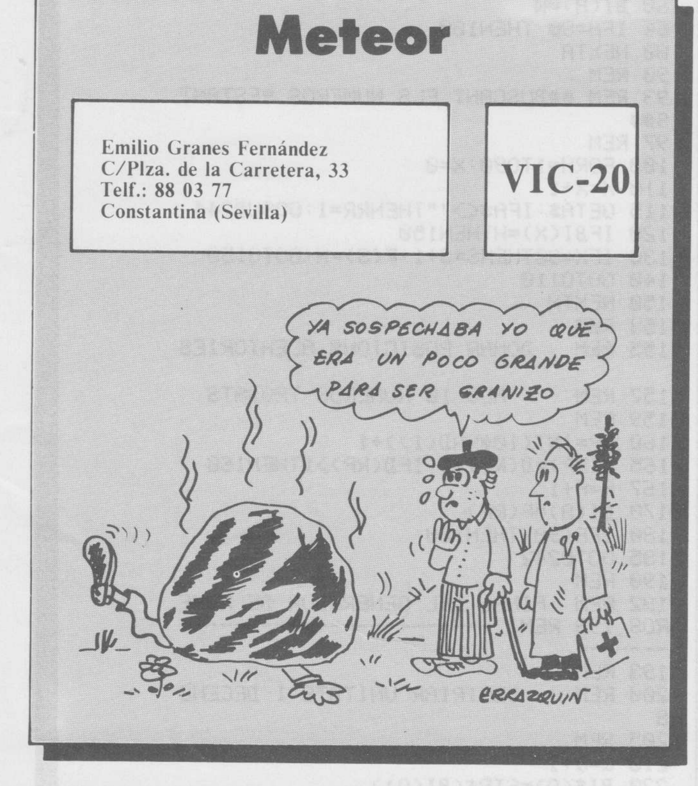
```
0 REM**CALCULO DE LAS FOT-FMU-FMP**
1 POKE53280,15:POKE53281,15:C$="[YEL]
[COMM+][COMM7]":GOSUB3000:GOSUB4000
18 PRINT"[CLR]"TAB(18)"[COMM@]"
19 PRINT"QUE[SPC]ESTACION[SPC]DEL[SPC]
ANO[SPC]DESEA"
20 PRINT"[2CRSRD]"TAB(14)"[RVSON]1-[RVSOFF]
[SPC][YEL]PRIMAVERA[COMM7]"
30 PRINT:PRINTTAB(14)"[RVSON]2-[RVSOFF]
[SPC][RED]VERANO[COMM7]"
35 PRINTTAB(20)"[COMM5][COMM@][COMM7]
"
40 PRINTTAB(14)"[RVSON]3-[RVSOFF][SPC]
[COMM5]OTONO[COMM7]"
50 PRINT:PRINTTAB(14)"[RVSON]4-[RVSOFF]
[SPC][BLK]INVIERNO[COMM7]"
60 PRINT:INPUTE
65 IFE<1ORE>4THENPRINT"[HOM][10CRSRD]
":GOTO60
100 PRINTTAB(34)"[COMM@]":PRINTTAB(5
)"INTRODUZCA[SPC]HORA[SPC]GMT,DIA,ME
S,ANO"
102 PRINTTAB(6)"DESDE[SPC]LAS[SPC]00
.00[SPC]A[SPC]LAS[SPC]26.00"
104 INPUTH,D,M,A
106 IFH<00.00ORH>26.00THENPRINT"[HOM]
[14CRSRD]":GOTO102
110 IFH>=02.00ORH<=14.00THEN150
120 MHZ=2*(27-H)
150 MHZ=2*(H-1)
500 IFE=1THEN540
510 IFE=2THENMHZ=MHZ+2
520 IFE=3THEN540
530 IFE=4THENMHZ=MHZ-2
540 FU=MHZ*30/100
550 FP=MHZ*50/100
600 PRINT"[CLR]":A$="***EL[SPC]DIA[16SPC]
A[SPC]LAS[7SPC]***"
604 PRINT"[HOM][5CRSRD]"A$
606 PRINT"[HOM][5CRSRD]"TAB(9)D"-"M"
-"A
608 PRINT"[HOM][5CRSRD]"TAB(30)H"GMT
":GOSUB6000
610 PRINT:PRINT"[3CRSRD][5CRSRR]LA[SPC]
FOT[SPC]APROXIMADA[SPC]ES:"MHZ"MHZ"
620 FU=MHZ+FU
630 FP=MHZ+FP
640 PRINT:PRINT"[5CRSRR]''[SPC]FMU[5SPC]
''[5SPC]'':"FU"MHZ"
650 PRINT:PRINT"[5CRSRR]''[SPC]FMP[5SPC]
''[5SPC]'':"FP"MHZ"
660 PRINT"[3CRSRD][7CRSRR]DESEA[SPC]
OTRO[SPC]CALCULO[SPC](S/N)":POKE198,
1:WAIT198,1
670 PRINT"[HOM][19CRSRD]"TAB(31):INP
UTOC$
680 IFOC$="N"THENEND
690 GOTO18
700 END
2999 REM RUTINA MARCO
3000 PRINT"[CLR]";:FORI=1TO39:PRINTC
$;:NEXT:PRINT"[CRSRU]"
3010 FORI=1TO20:PRINTC$TAB(38)C$:NEX
TI
3020 FORI=1TO39:PRINTC$;:NEXTI:PRINT
"[HOM]"
3030 RETURN
4000 A$="ESTE[SPC]PROGRAMA[SPC]CALCU
LA:":GOSUB5000:PRINT"[2CRSRD]"
4010 A$="LA[SPC]FRECUENCIA[SPC]OPTIM
A[SPC]DE[SPC]TRABAJO(FOT)":GOSUB5000
:PRINT"[2CRSRD]"
4020 A$="LA[SPC]FRECUENCIA[SPC]MAXIM
A[SPC]UTILIZABLE(FMU)":GOSUB5000:PRI
NT"[2CRSRD]"
4030 A$="LA[SPC]FRECUENCIA[SPC]MAXIM
A[SPC]POSIBLE[3SPC](FMP)":GOSUB5000:
PRINT"[2CRSRD]"
4040 A$="POR[SPC][YEL]LOMBAR[SPC](EA
1CMC)[COMM7]":GOSUB5000:PRINT"[2CRSRD]
"
4050 A$="JUNIO[SPC]1984":GOSUB5000:P
RINT"[2CRSRD]"
4060 A$="PULSE[SPC][RVSON]S[RVSOFF][SPC]
PARA[SPC]CONTINUAR":GOSUB5000
4070 GETP$:IFP$<>"S"THEN4070
```

```
4080 GOTO18
4090 END
4999 REM SUPER PRINT
5000 A=LEN(A$):IFA/2<>INT(A/2)THENA$
="[SPC]"+A$:GOTO5000
5010 FORI=1TOA/2
5020 PRINTTAB(20-I);LEFT$(A$,I);RIGH
T$(A$,I)
5030 PRINT"[CRSRU]";
5040 FORT=1TO200:NEXTT
5050 NEXTI
5060 RETURN
5999 REM SUBRAYADO
6000 L=40:IFLEN(A$)<40THENL=LEN(A$)
6020 IFL=40THENPRINT"[CRSRU]";
6030 FORI=1TOL
6040 PRINTCHR$(163);
6050 FORT=1TO100:NEXTT
6060 NEXTI
6070 A$=RIGHT$(A$,LEN(A$)-L)
6080 IFA$<>""THEN6000
6090 RETURN
```

Meteor

Emilio Granes Fernández
C/Plza. de la Carretera, 33
Telf.: 88 03 77
Constantina (Sevilla)

VIC-20



METEOR consiste en eliminar meteoritos (asteriscos) en minuto y medio, procurando que no te alcancen con sus rayos lásers. Tenéis que dirigirlo con las teclas del cursor para mover la nave y space para el disparo.

Basta de rollos, ahora paso a describiros por encima el programa.

Líneas 1-6: Definen el carácter de la nave.

Líneas 10-20: Reloj a cero. Color de marco y pantalla. Variables que intervienen en el "jueguecito".

Línea 35: Random para situar el primer meteorito del juego.

Línea 36: Poke de visualización para el carácter de la nave.

Línea 37: Compara el valor del reloj hasta uno mayor o igual que 90 sgs., saltando a la 209.

Líneas 50-70: Teclas de movimiento y disparo de la nave.

Líneas 209-220: Mensaje 1 de fin de programa.

Líneas 400-430: Crea el rastro y sonidos de las bombas de los meteoritos, borrándolos después.

Líneas 900-920: Mensaje 2 de fin de programa al ser tocada la nave.

Líneas 1000-1020: (gosub 1000) Crean el efecto de disparo de la nave, así como la explosión del meteorito tocado. Aquí se encuentra el contador de asteriscos destruidos.

Líneas 2000-2010: (gosub 2000) Crea el estallido del asterisco decreciendo el volumen.

```
1 POKE36874,0:POKE36875,0:POKE36876,
0:POKE36877,0:POKE36878,15
2 POKE52,28:POKE56,28:CLR
3 FORX=7168TO7679:POKEX,PEEK(X+25600
):NEXTX
4 POKE36869,255
5 FORL=7168TO7175:READD:POKEL,D:NEXT
L
6 DATA24,36,36,36,36,36,66,255
10 PRINT"[CLR]":TI$="000000"
15 POKE36879,104
20 P=8152:S2=36878:POKES2,15:S1=3687
6:R=36877:V=36879
35 D=INT(RND(0)*506)+7680:POKED,42
36 POKEP,0
37 IFTI$>="000090"THEN209
40 GETA$:IFA$=""THEN40
50 IFA$="[CRSRR]"THENP=P+1:POKEP-1,3
2:POKEP,0
55 IFP=8141THENP=8142:POKEP-1,32:POK
EP,0:GOTO37
60 IFA$="[CRSRD]"THENP=P-1:POKEP+1,3
2:POKEP,0
65 IFP=8164THENP=8163:POKEP+1,32:POK
EP,0:GOTO37
70 IFA$="[SPC]"THENGOSUB1000
200 GOTO37
209 PRINTCHR$(158)
210 PRINT"[CLR]":PRINTTI$;"[CRSRR]SG
S.":POKES1,200:FORF=15TO0STEP-1:POKE
S2,F:FORH=1TO100:NEXT:NEXT
215 PRINT"[2CRSRD][3CRSRR]SE[SPC]TE[SPC]
ACABO[SPC]EL[CRSRR]COM-[CRSRD]BUSTIB
LE;PERO[SPC]DESINTE-[CRSRD]GRASTE[SPC]
";A;"[CRSRR]METEORITOS"
216 INPUT"[3CRSRD][RVSON]OTRA[SPC]MA
S[RVSOFF]";B$
217 IFB$="S"THENPRINT"[CLR]":RUN
218 IFB$<>"S"THENPRINT"[CLR]":END
220 END
400 FORJ=1TO22:D=D+22:POKED,39:NEXTJ
:FORL=230TO175STEP-3:POKES1,L:FORM=1
TO20
410 NEXTM:NEXTL:POKES1,0
420 IFPEEK(D-66)=PEEK(P)THENPOKER,20
0:FORL=15TO0STEP-.05:POKES2,L:NEXT:G
OTO900
425 POKER,0:POKES2,15
427 FORJ=1TO21:D=D-22:POKED,32:NEXT:
D=D-22
430 RETURN
900 PRINT"[CLR]":PRINTCHR$(5):PRINT"
[HOM][5CRSRD]LO[SPC]SIENTO,METEOR[SPC]
GANA."
910 PRINT"[CRSRD]TIME:"TI$;"[CRSRR]S
GS.""[CRSRD]":PRINT"[CRSRD]METEORITO
S:[SPC]";A
920 GOTO216
1000 FORT=1TO21:P=P-22:POKEP,46:NEXT

1002 FORL=1TO5:FORM=250TO240STEP-3
1003 POKES1,M:NEXTM
1004 FORM=240TO250
1005 POKES1,M:NEXTM
1006 POKES1,0:NEXTL
1010 FORT=1TO20:POKEP,32:P=P+22:POKE
P,32:NEXT:P=P+22
1011 IFPEEK(P-22)=PEEK(D)THENPOKEV,1
4:FORJ=1TO250:NEXT:POKEV,104:POKER,1
40:GOSUB2000
1012 D=INT(RND(0)*38)+7724:POKED,42:
A=A+1:GOSUB400
1020 RETURN
2000 FORW=15TO0STEP-3:POKES2,W:FORT=
1TO250:NEXT:NEXT:POKER,0:POKES2,15
2010 RETURN
```

Bingo comput

Xavier Prat i Burdo
Raval de Sta. Eulalia, s/n.
Santa María D'Olo
Barcelona

C-64



Es un programa largo, aunque sencillo. Las subrutinas para representar los números en pantalla, ocupan gran parte del programa.

Las "extracciones" de los 90 números se hacen independientemente de su representación en pantalla. Primero se generan 80 números de los 90 disponibles. Después el programa va comprobando los números que faltan, y les asigna posiciones aleatorias.

No se tiene que esperar a que se hayan "extraído" todos los números para representarlos en pantalla. Sino que puedes empezar a representar los números en pantalla, y simultáneamente se irán sacando internamente los restantes números. Hay tiempo sobrado para que se acaben de extraer todos los números antes que tú de representarlos (a menos que mantengas siempre los espacios pulsados, saliendo así continuamente los números de la pantalla sin tiempo para jugar).

La tecla de función F7 está disponible para cuando se haya completado una línea o se haya completado todo el cartón. De esta forma se representa un tablero con todos los números sacados hasta el momento. Estos números están ordenados para la mejor localización de los mismos. Si después de comprobar un "BINGO", éste fuera correcto, F8 empezará automáticamente otra partida. En caso contrario F7 proseguirá el desarrollo del programa.

A continuación expongo un pequeño comentario de las líneas:

0- 80	Estas líneas extraen 80 números no repetidos entre el 1 y 90 y los guarda en una matriz.
90-150	En esta parte de programa, se buscan todos los 10 números que faltan por extraer.
153-180	Asignación de posiciones aleatorias a los 10 números anteriores.
210-239	Deshace el número en unidades y decenas para su próxima representación a gran tamaño en pantalla.
242-246	Espera para dibujar el número en pantalla; o representar el tablero.
260-270	Dibuja el número en pantalla.
278-287	Cuando la partida no se ha desarrollado como es debido, actúan estas líneas.
420-980	Subrutinas de gran tamaño de los 10 números (0 al 10).
2000-2080	Posicionado clasificado de todos los números salidos hasta el momento.
5000-5002	Subrutina para colocar el cursor en un punto determinado de la pantalla.
9990-10090	Subrutina para el dibujo del tablero.
25000-25020	Subrutina para imprimir el marco para la salida de los números en pantalla.

```
3 POKE53280,0:POKE53281,0
5 DIMBI(90),D(10),F(10),BI$(90)
10 PRINT"[WHT][CLR][12CRSRD][SPC]###
######[SPC]ESPERE´T[SPC]UN[SPC]MOMEN
T[SPC]########"
13 REM
15 REM *** GENERADOR DE 80 NUMEROS *
**
17 REM
20 FORA=1TO90:P=P+1
27 IFP=60THENP=0:GOSUB25000:RR=1:GOS
UB200:N=1
30 W=INT(89*RND(1))+1
35 IFN=0THEN40
39 GETA$:IFA$=CHR$(136)ORA$=CHR$(32)
THENRR=1:GOSUB244
40 FORY=1TOA:IFBI(Y)=WTHEN27
50 NEXTY
60 BI(A)=W
66 IFA=80 THEN100
80 NEXTA
90 REM
93 REM ##BUSCANT ELS NUMEROS RESTANT
S##
97 REM
100 FORH=1TO90:X=0
110 X=X+1
115 GETA$:IFA$<>""THENRR=1:GOSUB244
120 IFBI(X)=HTHEN150
130 IFX=80THENS=S+1:F(S)=H:GOTO150
140 GOTO110
150 NEXTH
153 REM
155 REM   DONAR POSICIONS ALEATORIES

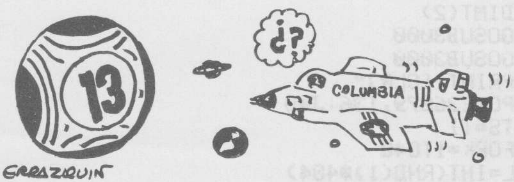
157 REM      ALS 10 NUMEROS TROBATS
159 REM
160 RP=INT(10*RND(1))+1
165 D(RP)=D(RP)+1:IFD(RP)>1THEN160
167 A=A+1
170 BI(A)=F(RP)
180 IFA<90 THEN160
185 GOTO242
190 REM
192 REM  FINAL DEL GENERADOR DE NUME
ROS 193 REM  ----------------------
-------
193 REM
200 REM    DESTRIAR UNITATS I DECENE
S
205 REM
210 Q=Q+1
220 BI$(Q)=STR$(BI(Q))
223 IFVAL(BI$(Q))<10THENBI$(Q)="[SPC]
"+BI$(Q)
230 ES$=LEFT$(BI$(Q),2):IFES$=""THEN
ES$="0"
235 DR$=RIGHT$(BI$(Q),1)
237 PRINT"[HOM][23CRSRD][SPC][RVSON]
**[SPC]ESPAI[RVSOFF][SPC]CANVIA[SPC]
NUMERO[2SPC][RVSON]**[SPC]F7[RVSOFF]
[SPC]TAULER
238 POKE198,0:REM BUIDAR BUFFER
239 IFRR=1THENRR=0:RETURN
242 GETA$:IFA$=""THEN242
244 IFA$=CHR$(136)THEN2000
246 IFA$<>CHR$(32)THEN239
250 PRINT"[HOM][CRSRD]":SP=-5
253 REM
255 REM   POSAR EL NUMERO EN PANTALL
A
257 REM
260 ONVAL(ES$)+1GOSUB420,600,676,705
```

```
,750,809,855,881,912,955
265 SP=12:PRINT"[HOM][YEL][CRSRD]"
270 ONVAL(DR$)+1GOSUB420,600,676,705
,750,809,855,881,912,955
272 IFQ=ATHENQ=Q+1:GOTO277
275 GOTO210
277 IFA=90THEN290
278 REM *** SI NO S'HAN GENERAT TOTS
 ELS  NUMEROS ES QUE NO S'HA JUGAT
 BE  ***
279 PRINT"[CLR][11CRSRD][5CRSRR]QUE[SPC]
NO[SPC]SAPS[SPC]JUGAR[SPC]AL[SPC]BIN
GO[SPC]?":FORM=1TO3000:NEXTM
280 PRINT"[3CRSRD][3SPC]APRETA[SPC]U
NA[SPC]TECLA[SPC]PER[SPC]TORNAR[SPC]
A[SPC]JUGAR":POKE198,0
285 GETY$:IFY$=""THEN285
287 RUN
289 REM *** FINAL DE LA PARTIDA ***
290 GETTT$:IFTT$=""THEN290
300 IFTT$=CHR$(136)ORTT$=CHR$(32)THE
N2000
310 GOTO290
420 REM *** NUMERO CERO ***
440 PRINTTAB(10+SP)"[RVSON][11SPC][RVSOFF]
"
450 PRINTTAB(10+SP)"[RVSON][11SPC][RVSOFF]
"
460 PRINTTAB(10+SP)"[RVSON][11SPC][RVSOFF]
"
465 PRINTTAB(10+SP)"[RVSON][3SPC][RVSOFF]
[5SPC][RVSON][3SPC][RVSOFF]"
470 PRINTTAB(10+SP)"[RVSON][3SPC][RVSOFF]
[5SPC][RVSON][3SPC][RVSOFF]"
472 PRINTTAB(10+SP)"[RVSON][3SPC][RVSOFF]
[5SPC][RVSON][3SPC][RVSOFF]"
475 PRINTTAB(10+SP)"[RVSON][3SPC][RVSOFF]
[5SPC][RVSON][3SPC][RVSOFF]"
480 PRINTTAB(10+SP)"[RVSON][3SPC][RVSOFF]
[5SPC][RVSON][3SPC][RVSOFF]"
485 PRINTTAB(10+SP)"[RVSON][3SPC][RVSOFF]
[5SPC][RVSON][3SPC][RVSOFF]"
490 PRINTTAB(10+SP)"[RVSON][3SPC][RVSOFF]
[5SPC][RVSON][3SPC][RVSOFF]"
495 PRINTTAB(10+SP)"[RVSON][3SPC][RVSOFF]
[5SPC][RVSON][3SPC][RVSOFF]"
500 PRINTTAB(10+SP)"[RVSON][3SPC][RVSOFF]
[5SPC][RVSON][3SPC][RVSOFF]"
515 PRINTTAB(10+SP)"[RVSON][3SPC][RVSOFF]
[5SPC][RVSON][3SPC][RVSOFF]"
520 PRINTTAB(10+SP)"[RVSON][3SPC][RVSOFF]
[5SPC][RVSON][3SPC][RVSOFF]"
525 PRINTTAB(10+SP)"[RVSON][3SPC][RVSOFF]
[5SPC][RVSON][3SPC][RVSOFF]"
530 PRINTTAB(10+SP)"[RVSON][11SPC][RVSOFF]
"
535 PRINTTAB(10+SP)"[RVSON][11SPC][RVSOFF]
"
540 PRINTTAB(10+SP)"[RVSON][11SPC][RVSOFF]
"
550 RETURN
600 REM*** SUBRUTINA NUMERO 1 ***
613 PRINTTAB(10+SP)"[5SPC][RVSON][3SPC]
[RVSOFF][4SPC]"
615 PRINTTAB(10+SP)"[4SPC][RVSON][4SPC]
[RVSOFF][4SPC]"
616 PRINTTAB(10+SP)"[3SPC][RVSON][5SPC]
[RVSOFF][4SPC]"
617 PRINTTAB(10+SP)"[2SPC][RVSON][6SPC]
[RVSOFF][4SPC]"
619 PRINTTAB(10+SP)"[2SPC][RVSON][6SPC]
[RVSOFF][4SPC]"
620 PRINTTAB(10+SP)"[5SPC][RVSON][3SPC]
[RVSOFF][4SPC]"
622 PRINTTAB(10+SP)"[5SPC][RVSON][3SPC]
[RVSOFF][4SPC]"
625 PRINTTAB(10+SP)"[5SPC][RVSON][3SPC]
[RVSOFF][4SPC]"
627 PRINTTAB(10+SP)"[5SPC][RVSON][3SPC]
[RVSOFF][4SPC]"
630 PRINTTAB(10+SP)"[5SPC][RVSON][3SPC]
[RVSOFF][4SPC]"
633 PRINTTAB(10+SP)"[5SPC][RVSON][3SPC]
[RVSOFF][4SPC]"
635 PRINTTAB(10+SP)"[5SPC][RVSON][3SPC]
[RVSOFF][4SPC]"
637 PRINTTAB(10+SP)"[5SPC][RVSON][3SPC]
[RVSOFF][4SPC]"
640 PRINTTAB(10+SP)"[5SPC][RVSON][3SPC]
[RVSOFF][4SPC]"
643 PRINTTAB(10+SP)"[5SPC][RVSON][3SPC]
[RVSOFF][4SPC]"
645 PRINTTAB(10+SP)"[5SPC][RVSON][3SPC]
[RVSOFF][4SPC]"
650 PRINTTAB(10+SP)"[5SPC][RVSON][3SPC]
[RVSOFF][4SPC]"
655 PRINTTAB(10+SP)"[5SPC][RVSON][3SPC]
[RVSOFF][4SPC]"
660 RETURN
675 REM *** SUBRUTINA NUMERO 2 ***
676 PRINTTAB(10+SP)"[RVSON][11SPC][RVSOFF]
"
677 PRINTTAB(10+SP)"[RVSON][11SPC][RVSOFF]
"
678 PRINTTAB(10+SP)"[RVSON][11SPC][RVSOFF]
"
679 PRINTTAB(10+SP)"[RVSON][3SPC][RVSOFF]
[5SPC][RVSON][3SPC][RVSOFF]"
680 PRINTTAB(10+SP)"[RVSON][3SPC][RVSOFF]
[5SPC][RVSON][3SPC][RVSOFF]"
681 PRINTTAB(10+SP)"[8SPC][RVSON][3SPC]
[RVSOFF][2SPC]"
682 PRINTTAB(10+SP)"[7SPC][RVSON][4SPC]
[RVSOFF][2SPC]"
683 PRINTTAB(10+SP)"[6SPC][RVSON][4SPC]
[RVSOFF][3SPC]"
684 PRINTTAB(10+SP)"[5SPC][RVSON][4SPC]
[RVSOFF][4SPC]"
685 PRINTTAB(10+SP)"[4SPC][RVSON][4SPC]
[RVSOFF][5SPC]"
686 PRINTTAB(10+SP)"[3SPC][RVSON][4SPC]
[RVSOFF][6SPC]"
687 PRINTTAB(10+SP)"[2SPC][RVSON][4SPC]
[RVSOFF][7SPC]"
688 PRINTTAB(10+SP)"[SPC][RVSON][4SPC]
[RVSOFF][8SPC]"
689 PRINTTAB(10+SP)"[RVSON][4SPC][RVSOFF]
[9SPC]"
691 PRINTTAB(10+SP)"[RVSON][3SPC][RVSOFF]
[11SPC]"
692 PRINTTAB(10+SP)"[RVSON][11SPC][RVSOFF]
"
693 PRINTTAB(10+SP)"[RVSON][11SPC][RVSOFF]
"
694 PRINTTAB(10+SP)"[RVSON][11SPC][RVSOFF]
"
695 RETURN
705 REM *** SUBRUTINA NUMERO 3 ***
708 PRINTTAB(10+SP)"[RVSON][11SPC][RVSOFF]
"
709 PRINTTAB(10+SP)"[RVSON][11SPC][RVSOFF]
"
710 PRINTTAB(10+SP)"[RVSON][11SPC][RVSOFF]
"
711 PRINTTAB(10+SP)"[8SPC][RVSON][3SPC]
[RVSOFF]"
712 PRINTTAB(10+SP)"[8SPC][RVSON][3SPC]
[RVSOFF]"
713 PRINTTAB(10+SP)"[8SPC][RVSON][3SPC]
[RVSOFF]"
714 PRINTTAB(10+SP)"[8SPC][RVSON][3SPC]
[RVSOFF]"
715 PRINTTAB(10+SP)"[5SPC][RVSON][6SPC]
[RVSOFF]"
716 PRINTTAB(10+SP)"[5SPC][RVSON][6SPC]
[RVSOFF]"
717 PRINTTAB(10+SP)"[5SPC][RVSON][6SPC]
[RVSOFF]"
718 PRINTTAB(10+SP)"[5SPC][RVSON][6SPC]
[RVSOFF]"
```

```
719 PRINTTAB(10+SP)"[8SPC][RVSON][3SPC]
[RVSOFF]"
720 PRINTTAB(10+SP)"[8SPC][RVSON][3SPC]
[RVSOFF]"
721 PRINTTAB(10+SP)"[8SPC][RVSON][3SPC]
[RVSOFF]"
722 PRINTTAB(10+SP)"[8SPC][RVSON][3SPC]
[RVSOFF]"
723 PRINTTAB(10+SP)"[RVSON][11SPC][RVSOFF]
"
724 PRINTTAB(10+SP)"[RVSON][11SPC][RVSOFF]
"
725 PRINTTAB(10+SP)"[RVSON][11SPC][RVSOFF]
"
730 RETURN
750 REM *** SUBRUTINA NUMERO 4 ***
755 PRINTTAB(10+SP)"[RVSON][3SPC][RVSOFF]
[5SPC][RVSON][3SPC][RVSOFF]"
756 PRINTTAB(10+SP)"[RVSON][3SPC][RVSOFF]
[5SPC][RVSON][3SPC][RVSOFF]"
757 PRINTTAB(10+SP)"[RVSON][3SPC][RVSOFF]
[5SPC][RVSON][3SPC][RVSOFF]"
758 PRINTTAB(10+SP)"[RVSON][3SPC][RVSOFF]
[5SPC][RVSON][3SPC][RVSOFF]"
759 PRINTTAB(10+SP)"[RVSON][3SPC][RVSOFF]
[5SPC][RVSON][3SPC][RVSOFF]"
760 PRINTTAB(10+SP)"[RVSON][3SPC][RVSOFF]
[5SPC][RVSON][3SPC][RVSOFF]"
768 PRINTTAB(10+SP)"[RVSON][3SPC][RVSOFF]
[5SPC][RVSON][3SPC][RVSOFF]"
770 PRINTTAB(10+SP)"[RVSON][11SPC][RVSOFF]
"
771 PRINTTAB(10+SP)"[RVSON][11SPC][RVSOFF]
"
772 PRINTTAB(10+SP)"[RVSON][11SPC][RVSOFF]
"
774 PRINTTAB(10+SP)"[8SPC][RVSON][3SPC]
[RVSOFF]"
775 PRINTTAB(10+SP)"[8SPC][RVSON][3SPC]
[RVSOFF]"
776 PRINTTAB(10+SP)"[8SPC][RVSON][3SPC]
[RVSOFF]"
777 PRINTTAB(10+SP)"[8SPC][RVSON][3SPC]
[RVSOFF]"
778 PRINTTAB(10+SP)"[8SPC][RVSON][3SPC]
[RVSOFF]"
779 PRINTTAB(10+SP)"[8SPC][RVSON][3SPC]
[RVSOFF]"
780 PRINTTAB(10+SP)"[8SPC][RVSON][3SPC]
[RVSOFF]"
781 PRINTTAB(10+SP)"[8SPC][RVSON][3SPC]
[RVSOFF]"
785 RETURN
800 REM *** SUBRUTINA NUMERO 5 ***
809 PRINTTAB(10+SP)"[RVSON][11SPC][RVSOFF]
"
810 PRINTTAB(10+SP)"[RVSON][11SPC][RVSOFF]
"
811 PRINTTAB(10+SP)"[RVSON][11SPC][RVSOFF]
"
812 PRINTTAB(10+SP)"[RVSON][3SPC][RVSOFF]
[8SPC]"
813 PRINTTAB(10+SP)"[RVSON][3SPC][RVSOFF]
[8SPC]"
814 PRINTTAB(10+SP)"[RVSON][3SPC][RVSOFF]
[8SPC]"
815 PRINTTAB(10+SP)"[RVSON][3SPC][RVSOFF]
[8SPC]"
816 PRINTTAB(10+SP)"[RVSON][3SPC][RVSOFF]
[8SPC]"
817 PRINTTAB(10+SP)"[RVSON][11SPC][RVSOFF]
"
818 PRINTTAB(10+SP)"[RVSON][11SPC][RVSOFF]
"
819 PRINTTAB(10+SP)"[RVSON][11SPC][RVSOFF]
"
820 PRINTTAB(10+SP)"[8SPC][RVSON][3SPC]
[RVSOFF]"
821 PRINTTAB(10+SP)"[8SPC][RVSON][3SPC]
[RVSOFF]"
822 PRINTTAB(10+SP)"[8SPC][RVSON][3SPC]
[RVSOFF]"
823 PRINTTAB(10+SP)"[8SPC][RVSON][3SPC]
[RVSOFF]"
824 PRINTTAB(10+SP)"[RVSON][11SPC][RVSOFF]
"
825 PRINTTAB(10+SP)"[RVSON][11SPC][RVSOFF]
"
827 PRINTTAB(10+SP)"[RVSON][11SPC][RVSOFF]
"
830 RETURN
850 REM *** SUBRUTINA NUMERO 6 ***
855 PRINTTAB(10+SP)"[RVSON][11SPC][RVSOFF]
"
856 PRINTTAB(10+SP)"[RVSON][11SPC][RVSOFF]
"
857 PRINTTAB(10+SP)"[RVSON][11SPC][RVSOFF]
"
858 PRINTTAB(10+SP)"[RVSON][3SPC][RVSOFF]
[8SPC]"
859 PRINTTAB(10+SP)"[RVSON][3SPC][RVSOFF]
[8SPC]"
860 PRINTTAB(10+SP)"[RVSON][3SPC][RVSOFF]
[8SPC]"
861 PRINTTAB(10+SP)"[RVSON][3SPC][RVSOFF]
[8SPC]"
862 PRINTTAB(10+SP)"[RVSON][3SPC][RVSOFF]
[8SPC]"
863 PRINTTAB(10+SP)"[RVSON][11SPC][RVSOFF]
"
864 PRINTTAB(10+SP)"[RVSON][11SPC][RVSOFF]
"
865 PRINTTAB(10+SP)"[RVSON][11SPC][RVSOFF]
"
866 PRINTTAB(10+SP)"[RVSON][3SPC][RVSOFF]
[5SPC][RVSON][3SPC][RVSOFF]"
867 PRINTTAB(10+SP)"[RVSON][3SPC][RVSOFF]
[5SPC][RVSON][3SPC][RVSOFF]"
868 PRINTTAB(10+SP)"[RVSON][3SPC][RVSOFF]
[5SPC][RVSON][3SPC][RVSOFF]"
869 PRINTTAB(10+SP)"[RVSON][3SPC][RVSOFF]
[5SPC][RVSON][3SPC][RVSOFF]"
870 PRINTTAB(10+SP)"[RVSON][11SPC][RVSOFF]
"
871 PRINTTAB(10+SP)"[RVSON][11SPC][RVSOFF]
"
872 PRINTTAB(10+SP)"[RVSON][11SPC][RVSOFF]
"
875 RETURN
879 REM *** SUBRUTINA NUMERO 7 ***
881 PRINTTAB(10+SP)"[RVSON][11SPC][RVSOFF]
"
882 PRINTTAB(10+SP)"[RVSON][11SPC][RVSOFF]
"
883 PRINTTAB(10+SP)"[RVSON][11SPC][RVSOFF]
"
884 PRINTTAB(10+SP)"[7SPC][RVSON][4SPC]
[RVSOFF]"
885 PRINTTAB(10+SP)"[6SPC][RVSON][4SPC]
[RVSOFF][SPC]"
886 PRINTTAB(10+SP)"[6SPC][RVSON][4SPC]
[RVSOFF][SPC]"
887 PRINTTAB(10+SP)"[5SPC][RVSON][4SPC]
[RVSOFF][2SPC]"
888 PRINTTAB(10+SP)"[5SPC][RVSON][4SPC]
[RVSOFF][2SPC]"
889 PRINTTAB(10+SP)"[4SPC][RVSON][4SPC]
[RVSOFF][3SPC]"
890 PRINTTAB(10+SP)"[4SPC][RVSON][4SPC]
[RVSOFF][3SPC]"
891 PRINTTAB(10+SP)"[3SPC][RVSON][4SPC]
[RVSOFF][4SPC]"
892 PRINTTAB(10+SP)"[3SPC][RVSON][4SPC]
[RVSOFF][4SPC]"
893 PRINTTAB(10+SP)"[2SPC][RVSON][4SPC]
[RVSOFF][5SPC]"
894 PRINTTAB(10+SP)"[2SPC][RVSON][4SPC]
[RVSOFF][5SPC]"
895 PRINTTAB(10+SP)"[SPC][RVSON][4SPC]
[RVSOFF][6SPC]"
```

```
896 PRINTTAB(10+SP)"[SPC][RVSON][4SPC]
[RVSOFF][6SPC]"
897 PRINTTAB(10+SP)"[RVSON][4SPC][RVSOFF]
[7SPC]"
898 PRINTTAB(10+SP)"[RVSON][4SPC][RVSOFF]
[7SPC]"
900 RETURN
910 REM *** SUBRUTINA NUMERO 8 ***
912 PRINTTAB(10+SP)"[RVSON][11SPC][RVSOFF]
"
913 PRINTTAB(10+SP)"[RVSON][11SPC][RVSOFF]
"
914 PRINTTAB(10+SP)"[RVSON][11SPC][RVSOFF]
"
915 PRINTTAB(10+SP)"[RVSON][3SPC][RVSOFF]
[5SPC][RVSON][3SPC][RVSOFF]"
916 PRINTTAB(10+SP)"[RVSON][3SPC][RVSOFF]
[5SPC][RVSON][3SPC][RVSOFF]"
917 PRINTTAB(10+SP)"[RVSON][3SPC][RVSOFF]
[5SPC][RVSON][3SPC][RVSOFF]"
918 PRINTTAB(10+SP)"[RVSON][3SPC][RVSOFF]
[5SPC][RVSON][3SPC][RVSOFF]"
919 PRINTTAB(10+SP)"[RVSON][11SPC][RVSOFF]
"
920 PRINTTAB(10+SP)"[RVSON][11SPC][RVSOFF]
"
921 PRINTTAB(10+SP)"[RVSON][11SPC][RVSOFF]
"
922 PRINTTAB(10+SP)"[RVSON][11SPC][RVSOFF]
"
923 PRINTTAB(10+SP)"[RVSON][3SPC][RVSOFF]
[5SPC][RVSON][3SPC][RVSOFF]"
924 PRINTTAB(10+SP)"[RVSON][3SPC][RVSOFF]
[5SPC][RVSON][3SPC][RVSOFF]"
925 PRINTTAB(10+SP)"[RVSON][3SPC][RVSOFF]
[5SPC][RVSON][3SPC][RVSOFF]"
926 PRINTTAB(10+SP)"[RVSON][3SPC][RVSOFF]
[5SPC][RVSON][3SPC][RVSOFF]"
927 PRINTTAB(10+SP)"[RVSON][11SPC][RVSOFF]
"
928 PRINTTAB(10+SP)"[RVSON][11SPC][RVSOFF]
"
929 PRINTTAB(10+SP)"[RVSON][11SPC][RVSOFF]
"
932 RETURN
950 REM *** SUBRUTINA NUMERO 9 ***
955 PRINTTAB(10+SP)"[RVSON][11SPC][RVSOFF]
"
956 PRINTTAB(10+SP)"[RVSON][11SPC][RVSOFF]
"
957 PRINTTAB(10+SP)"[RVSON][11SPC][RVSOFF]
"
958 PRINTTAB(10+SP)"[RVSON][3SPC][RVSOFF]
[5SPC][RVSON][3SPC][RVSOFF]"
959 PRINTTAB(10+SP)"[RVSON][3SPC][RVSOFF]
[5SPC][RVSON][3SPC][RVSOFF]"
960 PRINTTAB(10+SP)"[RVSON][3SPC][RVSOFF]
[5SPC][RVSON][3SPC][RVSOFF]"
961 PRINTTAB(10+SP)"[RVSON][3SPC][RVSOFF]
[5SPC][RVSON][3SPC][RVSOFF]"
962 PRINTTAB(10+SP)"[RVSON][11SPC][RVSOFF]
"
963 PRINTTAB(10+SP)"[RVSON][11SPC][RVSOFF]
"
964 PRINTTAB(10+SP)"[RVSON][11SPC][RVSOFF]
"
965 PRINTTAB(10+SP)"[8SPC][RVSON][3SPC]
[RVSOFF]"
966 PRINTTAB(10+SP)"[8SPC][RVSON][3SPC]
[RVSOFF]"
967 PRINTTAB(10+SP)"[8SPC][RVSON][3SPC]
[RVSOFF]"
968 PRINTTAB(10+SP)"[8SPC][RVSON][3SPC]
[RVSOFF]"
969 PRINTTAB(10+SP)"[8SPC][RVSON][3SPC]
[RVSOFF]"
970 PRINTTAB(10+SP)"[8SPC][RVSON][3SPC]
[RVSOFF]"
972 PRINTTAB(10+SP)"[8SPC][RVSON][3SPC]
[RVSOFF]"
973 PRINTTAB(10+SP)"[8SPC][RVSON][3SPC]
[RVSOFF]"
980 RETURN
2000 IFQ=1THEN239:REM CAP NUMERO TRE
T
2003 PRINT"[CLR]":GOSUB10000
2005 FORC=1TOQ-1
2010 ES$=LEFT$(BI$(C),2)
2020 DR$=RIGHT$(BI$(C),1)
2030 COL=VAL(ES$)*4+2:FIL=VAL(DR$)*2

2033 IFDR$="0"THENFIL=20:COL=COL-4
2035 IFQ=1THEN2050
2040 GOSUB 5000
2050 NEXTC
2055 PRINT"[HOM][23CRSRD][2SPC][RVSON]
**[SPC]F7[SPC][RVSOFF][SPC]CONTINUA[2SPC]
[RVSON]**[SPC]F8[SPC][RVSOFF][SPC]NO
VA[SPC]PARTIDA
2060 GETB$:IFB$=""THEN2060
2065 IFB$=CHR$(140)THENRUN
2067 IFQ=91THEN2060
2070 IFB$<>CHR$(136)THEN2060
2076 GOSUB25000
2080 Q=Q-1:ES$=LEFT$(BI$(Q),2):DR$=R
IGHT$(BI$(Q),1):GOTO250
5000 D$="[HOM][40CRSRD]"
5002 PRINTLEFT$(D$,FIL);TAB(COL) ;BI
(C):RETURN
9990 REM
9995 REM*** SUBRUTINA PER FER QUADRE
 ***
9997 REM
9999 FORT=1TO90:PRINTBI(T),     :NEX
T
10000 PRINT"[HOM][SPC][COMMM][3COMMY]
[SHIFTP][3COMMY][SHIFTP][3COMMY][SHIFTP]
[3COMMY][SHIFTP][3COMMY][SHIFTP][3COMMY]
[SHIFTP][3COMMY][SHIFTP][3COMMY][SHIFTP]
[3COMMY][SHIFTP]"
10030 FORT=0TO18
10040 PRINT"[SPC][COMMM][3SPC][COMMM]
[3SPC][COMMM][3SPC][COMMM][3SPC][COMMM]
[3SPC][COMMM][3SPC][COMMM][3SPC][COMMM]
[3SPC][COMMM][3SPC][COMMM][SPC]"
10050 NEXTT
10060 PRINT"[SPC][COMMM][3COMMP][SHIFT@]
[3COMMP][SHIFT@][3COMMP][SHIFT@][3COMMP]
[SHIFT@][3COMMP][SHIFT@][3COMMP][SHIFT@]
[3COMMP][SHIFT@][3COMMP][SHIFT@][3COMMP]
[SHIFT@][SPC]"
10064 PRINT"[SPC][COMMM][RVSON][SPC]
0[3SPC]10[2SPC]20[2SPC]30[2SPC]40[2SPC]
50[2SPC]60[2SPC]70[2SPC]80[SPC][RVSOFF]
"
10070 PRINT"[CRSRR][SPC][36COMMY][SPC]
"
10090 RETURN
24900 REM
24910 REM FORMACIO DEL MARC EN PANTA
LLA
24920 REM
25000 PRINT"[CLR][COMMM][36COMMT][COMMG]
"
25015 FORN=1TO20:PRINT"[COMMM]"TAB(3
7)"[COMMG]":NEXT
25020 PRINT"[COMMM][36COMM@][COMMG][SPC]
"
25030 RETURN
```



Ladrón

Francisco Belinchón Monjas
Getafe.
(Madrid)

VIC-20
Sin ampliación de memoria.



El principal objetivo es intentar robar la mayor cantidad de botines posibles, pero hay que tener cuidado con los policías y las múltiples farolas. Ahora paso a explicaros el programa.

- 1- 9 Creador de los nuevos caracteres.
- 50- 80 Coloca las farolas.
- 90 Crea los botines.
- 110- 169 Creación, movimiento y borrado de los policías. Además evalúa las situaciones cuando un policía se encuentra con un ladrón, un botín o una farola.
- 179- 780 Creación, movimiento y borrado del ladrón. Formula las situaciones en las cuales el ladrón se encuentra con un botín un policía o una farola.
- 2100-2211 Printa el final del juego.
- 3000-3085 Subrutina de comienzo del juego.
- 6000 Situación en donde se encuentran dos policías.

Para terminar os diré que tengo un VIC-20, 14 años de edad y os deseo que sigáis editando la revista por muchos años.

```
1 DATA28,12,12,254,199,221,39,97,0,1
6,31,18,90,110,36,0,24,60,24,56,124,
124,56,24
2 DATA124,68,78,68,64,64,64,224
3 POKE52,28:POKE56,28:CLR:FORU=7168T
O7679:POKEU,PEEK(U+25600):NEXTU
4 POKE650,128
5 POKE36869,255:T=31
6 FORI=7168+T*8TO7168+T*8+7:READD:PO
KEI,D:NEXTI
7 IFT<>30ANDT<>35ANDT<>37THENT=30:GO
TO6
8 IFT<>35ANDT<>37THENT=35:GOTO6
9 IFT<>37ANDT=31ORT=30ORT=35THENT=37
:GOTO6
11 DIMT(2)
12 GOSUB3000
13 GOSUB3000
14 PRINT"[CLR]"
15 POKE36879,136:J=0
22 TS=TI
50 FORK=1TO40
60 L=INT(RND(1)*484)
70 POKE7702+L,37:POKE38422+L,1
80 NEXTK
82 T(0)=0:T(1)=242:T(2)=484:Z=7680:X
=38400:C=36878:V=36876
90 FORT=1TO80:D=INT(RND(1)*484):POKE
7702+D,35:POKE38422+D,7:NEXTT
110 FORY=0TO2:POKEZ+T(Y),32:NEXTY
115 FORU=0TO2
120 IFA>T(U)ANDA-T(U)<=22THENT(U)=T(
U)+1
130 IFT(U)>AANDT(U)-A<=22THENT(U)=T(
U)-1
140 IFA>T(U)ANDA-T(U)>=22THENT(U)=T(
U)+22
150 IFT(U)>AANDT(U)-A>=22THENT(U)=T(
U)-22
160 IFPEEK(Z+T(U))=31THENPOKEC,15:PO
KEV,237:FORW=1TO900:NEXTW:POKEC,0:GO
SUB2100
162 IFPEEK(Z+T(U))=35THENPOKEZ+T(U),
35:POKEX+T(U),7:GOTO120
163 IFPEEK(Z+T(U))=37THENPOKEZ+T(U),
37:POKEX+T(U),1:GOTO120
164 POKEZ+T(U),30:POKEX+T(U),0
165 K=30
166 FORI=0TO2
167 IF(Z+T(I)+1)=KORPEEK(Z+T(I)-1)=K
ORPEEK(Z+T(I)+22)=KORPEEK(Z+T(I)-22)
=KTHENGOSUB6000
169 NEXTI,U
179 FORT=1TO4
180 GETA$
190 POKEZ+A,31:POKEX+A,2
200 IFA$="T"THENA=A-22:POKEZ+A+22,32

300 IFA$="F"THENA=A-1:POKEZ+A+1,32
600 IFA$="V"THENA=A+22:POKEZ+A-22,32

700 IFA$="G"THENA=A+1:POKEZ+A-1,32
720 IFPEEK(Z+A)=35THENJ=J+1:POKEC,15
:POKEKEV,205:FORU=1TO8:NEXTU:POKEC,0

730 IFPEEK(Z+A)=37ORPEEK(Z+A)=KTHENP
OKEC,9:POKEV,207:FORT=1TO900:NEXT:PO
KEC,0:GOTO2100
740 IFPEEK(Z+A)=37ORPEEK(Z+A)=30THEN
GOTO2100
750 NEXTT
780 GOTO110
1085 :60
2100 POKE36879,122
2110 PRINT"[CLR][2CRSRD][2CRSRR]BOTI
NES[SPC]ROBADOS=[SPC]";J
2200 PRINT"[2CRSRD][2CRSRR]TIEMPO[SPC]
EMPLEADO=[SPC]";INT((TI-TS)/60)
2205 IFJ>RTHENR=J
2206 PRINT"[RED][2CRSRD]EL[SPC]RECOR
D[SPC]ES[SPC]";R
2207 PRINT"[BLK][2CRSRD]TU[SPC]PUNTU
ACION[SPC]ES";J
2208 INPUT"[2CRSRD]QUIERE[SPC]JUGAR[SPC]
(S/N)";S$
2209 IFS$="N"THENGOTO2211
2210 GOTO13
2211 END
3000 PRINT"[CLR][2CRSRD][RVSON][RED]
LADRONES[SPC]Y[SPC]POLICIAS"
3010 PRINT"[3CRSRD][3CRSRR]MUEVE[SPC]
AL[SPC]LADRON[SPC]CON"
3020 PRINT"[CRSRD]'T'PARA[SPC]ARRIBA
"
3030 PRINT"[CRSRD]'F'[SPC]PARA[SPC]I
ZQUIERDA"
3040 PRINT"[CRSRD]'V'PARA[SPC]ABAJO"

3050 PRINT"[CRSRD]'G'PARA[SPC]DERECH
A"
3060 PRINT"[2CRSRD]PROCURA[SPC]COJER
[SPC]TODOS[SPC]LOS[SPC]BOTINES[SPC]S
IN[SPC]CHOCARTE[SPC]CON[SPC]LAS[SPC]
```

```
FAROLAS"
3070 PRINT"[CRSRD]CUIDADO[SPC]CON[SPC]
LA[SPC]POLI"
3075 GETA$:IFA$=""THEN3075
3080 RETURN
6000 POKEZ+T(I),32:T(I)=INT(RND(1)*5
06):RETURN
```

El ahorcado

José Francisco Torres Caro
Calle Apolo, 93
Telf.: (965) 71 00 24
Torrevieja (Alicante)

C-64

El programa que os envío es una versión del conocido juego del, AHORCADO, la idea la cogí de un programa de televisión que se llama "Leo contra todos", que ponen los jueves por la tarde. El juego es para dos jugadores, uno mete una palabra y el otro tiene que adivinarla diciendo letras, con el problema de que si fallas o agotas el tiempo establecido se dibuja un nuevo trazo del ahorcado. En total son nueve intentos erróneos al final de los cuales el programa te dice la palabra a adivinar y el juego finaliza.
Para el dibujo de la horca y del ahorcado he utilizado variables que son cadenas de espacios blancos inversos, es decir que no he utilizado para nada la memoria de pantalla ni la memoria de color.

```
5 PRINT"[CLR]":POKE53281,6:POKE53280
,6
6 GOTO1000
10 F=2:C=5:GOSUB10000:INPUT"ESCRIBE[SPC]
UNA[SPC]PALABRA";A$

12 PRINT"[CRSRU][43SPC]"
15 DIMA$(30):DIMP$(30):L2=1
20 FORN=1TOLEN(A$)
30 A$(N)=MID$(A$,N,1)
40 NEXTN
50 PRINT"[CRSRU]"

60 FORK=1TOLEN(A$)
70 PRINTTAB(1)"-";
80 NEXTK
90 I=LEN(A$)
100 F=6:C=3:GOSUB10000:PRINT"ESCRIBE
[SPC]UNA[SPC]LETRA"

110 E$="[WHT][RVSON][30SPC]"
120 PRINT"[10CRSRU]";E$
130 V=30
140 V=V-1
150 IFV=1THENPRINT"[CRSRU]TIEMPO[SPC]
AGOTADO":FORN=1TO1000:NEXTN:IFP=9THE
N2000
155 IFV=1THENV=30:PRINT"[HOM][19SPC]
":P=P+1:GOTO2000
160 PRINT"[CRSRU]"TAB(V);CHR$(32)
170 FORW=1TO250:NEXT W
180 GETB$:IFB$=""THEN200
190 L2=L2+1:GOTO210
200 GOTO140
210 IF P1>0 THEN 320
230 FORL=0TOLEN(A$)
240 IFB$=A$(L)THENI=I-1:IND=1:P1=P1+
1:P$(P1)=B$:GOTO280
250 NEXTL:IFIND=0THENP=P+1:GOTO2000
255 IND=0
260 F=18:C=L2:GOSUB10000:PRINTB$:IFP
=9THENPRINT"LO[SPC]SIENTO":GOTO500
270 GOTO100
280 F=4:C=L:GOSUB10000:PRINTB$:IFI=0
THEN310
290 GOTO250
310 PRINT"[4CRSRD]"TAB(11)"MUY[SPC]B
IEN":PRINTTAB(6)"[CRSRD]LO[SPC]HAS[SPC]
ENCONTRADO":END
320 FOR FO=1 TO P1
330 IF B$=P$(FO) THEN 260
340 NEXT FO:GOTO 230
500 PRINT"[CRSRD]LA[SPC]PALABRA[SPC]
ERA":PRINTTAB(16);A$:END
1000 PRINTTAB(10)"E[SPC]L[3SPC]A[SPC]
H[SPC]O[SPC]R[SPC]C[SPC]A[SPC]D[SPC]
O[SPC]"
1010 FORN=1TO2000:NEXT:PRINTTAB(10)"
[CRSRU][23SPC]"
1015 S$="[WHT][RVSON][8SPC]"
1020 PRINT"[3CRSRD]"TAB(25)S$
1030 PRINTTAB(25)"[WHT][RVSON][COMMH]
"
1035 PRINT"[3CRSRU]"
1040 L=L+1
1050 PRINTTAB(33)"[WHT][RVSON][SPC]"

1060 IFL=16THEN1080
1070 GOTO1040
1080 E$="[BLK][RVSON][13SPC][WHT]"
1090 H=26
1100 H=H-1
1110 PRINTTAB(H)E$
1120 IFH=21THEN10
1130 GOTO1100
2000 ONPGOTO2050,2150,2250,2350,2450
,2550,2650,2750,2850
2050 PRINT"[3CRSRD]"
2100 FORX=1TO2
2110 PRINTTAB(25)"[WHT][RVSON][2SPC]
"
2120 NEXTX:GOTO260
2150 PRINT"[5CRSRD]"
2200 PRINTTAB(24)"[BLK][RVSON][4COMMU]
[WHT]":GOTO260
2250 PRINT"[6CRSRD]"
2300 FORX=1TO4
2310 PRINTTAB(25)"[WHT][RVSON][2SPC]
"
2320 NEXT:GOTO260
2350 PRINT"[6CRSRD]"
2400 FORX=1TO4
2410 PRINTTAB(24)"[WHT][RVSON][COMML]
"
2420 NEXTX:GOTO260
```

```
2500 FORX=1TO4
2510 PRINTTAB(27)"[WHT][RVSON][COMMJ]
"
2520 NEXTX:GOTO260
2550 PRINT"[10CRSRD]"
2600 FORX=1TO3
2610 PRINTTAB(25)"[WHT][RVSON][COMML]
"
2620 NEXTX:GOTO260
2650 PRINT"[10CRSRD]"
2700 FORX=1TO3
2710 PRINTTAB(26)"[WHT][RVSON][COMMJ]
"
2720 NEXTX:GOTO260
2750 PRINT"[13CRSRD]"
2800 PRINTTAB(24)"[BLK][RVSON][COMMO]
[WHT]":GOTO260
2850 PRINT"[13CRSRD]"
2900 PRINTTAB(27)"[BLK][RVSON][COMMO]
[WHT]":GOTO260
10000 POKE781,F:POKE782,C:POKE783,0:
SYS65520:RETURN
```

Pesca submarina

Javier Martínez Reches
Carretera Vella, 6-8, Atº 3ª
Montcada Reixac (Barcelona)

VIC-20

Es una divertida aventura submarina, atractiva por la alta resolución de los caracteres y la combinación de sonidos y colores. Al ejecutar el programa aparece un mensaje de presentación durante unos segundos (mientras son copiados los nuevos caracteres), seguido del cual aparece en la pantalla un submarinista bastante bien definido ya que ocupa una extensión de once caracteres. Además del submarinista aparecen peces y erizos. Tu misión es dirigir al submarinista entre las diversas grutas submarinas que irán apareciendo al azar (hay hasta 7 cuevas diferentes) evitando chocar con los erizos y con las grutas. Cuando una parte delantera del submarinista choca con algo que no sea un pez la pantalla cambia de color y te informa de los puntos que llevas obtenidos hasta el momento. Los puntos los obtienes por cada erizo o pez alcanzado por su fusil submarino. Puedes chocar hasta 3 veces.

Los mandos del submarino son:
F1: arriba. F7: abajo. ESPACIO: arpón.
¡Buena pesca! Y hasta pronto.

```
1 REM PESCA SUBMARINA*JAVIER MARTINE
Z
25 PRINT"[CLR][RVSON][3SPC]PESCA[2SPC]
SUBMARINA[3SPC][RVSOFF]"
50 POKE52,28:POKE56,28:CLR
55 FORX=7168TO7679:POKEX,PEEK(X+2560
0):NEXTX
57 POKE36869,255
60 FORL=7168TO7279:READD:POKEL,D:NEX
TL
69 I=7680:U=36876:A=36877:E=36878:SU
=3
70 POKEE,15:POKE36879,110:PRINT"[CLR]
[HOM]"
90 X=6:Y=11
95 FORBU=1TO15
96 BU=INT(RND(1)*21)+1:BA=INT(RND(1)
*21)+1
97 POKEI+((22*BU)+BA),13
98 POKE38400+((22*BU)+BA),5
99 NEXTBU
100 FORJO=1TO10
101 JU=INT(RND(1)*21)+1:JA=INT(RND(1
)*21)+1
102 POKEI+((22*JU)+JA),12
103 POKE38400+((22*JU)+JA),3
105 NEXTJO
115 GETA$
117 IFA$="[F1]"THENR=3
120 IFA$="[F7]"THENR=5
123 IFA$="[SPC]"THEN500
130 POKEU,140:POKEI+X+22*Y,32
131 POKEI+X-1+22*Y,32
132 POKEI+X-2+22*Y,32
133 POKEI+X-3+22*Y,32
134 POKEI+X-4+22*Y,32
135 POKEI+X-5+22*Y,32
136 POKEI+(X-1)+22*(Y-1),32:POKEU,15
0
137 POKEI+(X-2)+22*(Y-1),32
138 POKEI+(X-3)+22*(Y-1),32
139 POKEI+(X-4)+22*(Y-1),32:POKEI+(X
-5)+22*(Y-1),32:X=X+1:POKEU,0
141 IFPEEK(I+X-1+22*Y)=11ORPEEK(I+X-
1+22*Y)=12THENSU=SU-1:GOTO1000
142 IFPEEK(I+X+22*Y)=11ORPEEK(I+X+22
*Y)=12THENSU=SU-1:GOTO1000
143 IFPEEK(I+X-1+22*(Y-1))=11ORPEEK(
I+X-1+22*(Y-1))=12THENSU=SU-1:GOTO10
00
144 IFR=3THENY=Y-1
145 IFR=5THENY=Y+1
146 IFX=28THENX=6:GOTO800
147 IFY=21THENY=20
148 IFY=0THENY=1
149 IFPEEK(I+X-2+22*(Y-1))=11THENSU=
SU-1:GOTO1000
150 POKEI+X+22*Y,10
151 POKEI+X-1+22*Y,9
152 POKEI+X-2+22*Y,8
153 POKEI+X-3+22*Y,7
154 POKEI+X-4+22*Y,6
155 POKEI+X-5+22*Y,5
156 POKEI+(X-1)+22*(Y-1),4
157 POKEI+(X-2)+22*(Y-1),3
158 POKEI+(X-3)+22*(Y-1),2
159 POKEI+(X-4)+22*(Y-1),1
160 POKEI+(X-5)+22*(Y-1),0
165 IFPEEK(I+X+22*Y)=11THENSU=SU-1:G
OTO1000
166 IFPEEK(I+X-1+22*(Y-1))=11THENSU=
SU-1:GOTO1000
170 GOTO115
500 J=X:POKEI+J+1+22*Y,32
505 FORK=15TO1STEP-1:J=J+1
506 POKEA,250:POKEE,K
508 IFPEEK(I+J+22*Y)=11THENPOKEE,15:
GOTO550
509 IFPEEK(I+J+22*Y)=13ORPEEK(I+J+22
```

```
*Y)=12THENPOKEE,15:GOTO580
510 POKEI+J+22*Y,45
515 POKEI+J+22*Y,32
520 NEXTK
525 POKEA,0:POKEE,15:GOTO115
550 POKEI+(J-1)+22*Y,32
551 POKEU,235
552 FORT=1TO50:NEXTT:POKEU,0:POKEA,0

553 GOTO115
580 POKEU,200
581 POKEI+(J-1)+22*Y,32
582 POKEI+J+22*Y,32
585 FORT=1TO100:NEXTT
590 POKEU,0:PUN=PUN+10:POKEA,0:GOTO1
15
800 OK=INT(RND(1)*7)+1
801 IFOK=2THEN70
802 IFOK=1THEN810
803 IFOK=3THEN830
804 IFOK=4THEN840
805 IFOK=5THEN870
806 IFOK=6THEN880
807 IFOK=7THEN890
810 PRINT"[CLR][RED]KK[CRSRD]KK[CRSRD]
KK[CRSRD]KK[CRSRU]KKKK[CRSRD]KK[CRSRD]
KKKKKK[CRSRU]KK"
820 PRINT"[17CRSRD]KK[CRSRU]KK[CRSRU]
KKKK[CRSRD]KKK[CRSRU]KK[CRSRU]K[CRSRD]
KKKKK[CRSRU]KKK":GOTO95
830 PRINT"[CLR][WHT]K[CRSRD]KK[CRSRD]
K[CRSRD]K[CRSRD]KK[CRSRD]K[CRSRD]KKK
[CRSRU]KK[CRSRD]KK[CRSRU]K[CRSRD]K[CRSRU]
KK[CRSRD]K[CRSRU]KK"
835 PRINT"[9CRSRD]K[CRSRU]K[CRSRU]K[CRSRU]
KKK[CRSRD]KKK[CRSRU]KK[CRSRU]KK[CRSRD]
KKK[CRSRU]K[CRSRU]K[CRSRD]KK[CRSRU]K
K":GOTO95
840 PRINT"[CLR][CYN]K[CRSRD]KK[CRSRD]
KK[CRSRD]KKKK[CRSRD]KKKK[CRSRD]K[CRSRD]
K[CRSRD]KKK[CRSRU]K[CRSRU]K[CRSRD]K"

845 PRINT"[13CRSRD]KKKKK[CRSRU]KKK[CRSRU]
K[CRSRU]K[CRSRU]K[CRSRU]K[CRSRU]K[CRSRD]
K[CRSRU]KKK[CRSRU]K[CRSRU]KKK":GOTO9
5
870 PRINT"[CLR][PUR]KK[CRSRD]KK[CRSRD]
KK[CRSRD]K[CRSRD]K[CRSRU]K[CRSRD]KK[CRSRD]
K[CRSRD]KK[CRSRD]K[CRSRD]KKK[CRSRD]K
[CRSRU]K[CRSRU]K[CRSRU]K"
875 PRINT"[10CRSRD]KKK[CRSRU]K[CRSRU]
K[CRSRU]KK[CRSRD]K[CRSRD]K[CRSRU]K[CRSRU]
K[CRSRD]K[CRSRU]K[CRSRU]KKK[CRSRU]K
K[CRSRD]KKK":GOTO95
880 PRINT"[CLR][GRN][CRSRD]K[CRSRD]K
[CRSRD]K[CRSRD]K[CRSRD]K[CRSRD]K[CRSRD]
K[CRSRD]K[CRSRD]KKK[CRSRU]K[CRSRU]K[CRSRD]
KKK[CRSRD]K[CRSRD]K[CRSRD]K[CRSRU]K[CRSRU]
K[CRSRU]K"
885 PRINT"[14CRSRD]K[CRSRU]K[CRSRU]K
KK[CRSRU]K[CRSRU]K[CRSRU]KK[CRSRU]K[CRSRD]
K[CRSRD]K[CRSRU]KKK[CRSRU]K[CRSRU]K[CRSRD]
K[CRSRD]KK[CRSRU]KK":GOTO95
890 PRINT"[CLR][YEL]K[CRSRD]K[CRSRD]
K[CRSRD]K[CRSRD]K[CRSRD]K[CRSRD][CRSRL]
KK[CRSRU]KKK[CRSRD]K[CRSRD]K[CRSRD]K
KKK[CRSRU]KKK[CRSRD]KK"
895 PRINT"[14CRSRD]K[CRSRU]K[CRSRU]K
[CRSRU]K[CRSRU]KKK[CRSRD]K[CRSRD]KK[CRSRU]
KKKK[CRSRU]K[CRSRD]KKK[CRSRU]K[CRSRD]
KK[CRSRU]K[CRSRU]K":GOTO95
1000 POKE36879,61:POKE36874,160:PRIN
T"[HOM][BLU][10CRSRD][6CRSRR]PUNTOS:
"PUN
1001 FORT=1TO3000:NEXT:POKE36874,0:I
FSU=0THENEND
1002 GOTO70
20000 DATA0,0,0,0,0,0,0,15,0,0,0,0,0
,0,0,248,0,0,0,0,0,0,63,127,0,0,0,0,
0,0,192,225
20001 DATA0,0,0,0,0,0,240,248,31,48,
64,143,31,48,64,128,255,15,7,255,255
,0,0,0
20002 DATA255,255,255,255,255,0,0,0,
255,255,255,255,255,3,1,0,248,240,22
4,255,130,176
20003 DATA224,192,0,0,0,255,0,0,0,0,
0,0,64,53,85,89,137,129,68,41,154,12
4
20004 DATA62,89,148,34,0,0,125,237,6
3,255,121,0
```

Programación del nim

Antonio Luis Climent Albaladejo
Comuneros, 15
30003 Murcia

C-64



El juego consiste en ir retirando fichas de varios montones, de manera que en cada jugada es posible tomar una, dos o incluso todas las de uno de ellos. Gana el jugador que retira la/s última/s ficha/s.

El armazón del programa lo forman unas pocas instrucciones más.

Los dibujos se realizan con gráficos de baja resolución (figura 2).

El resto de líneas son el adorno de todo programa que, como éste, pretenda ser lúdico.

Las instrucciones 50-70 y la subrutina 10100 definen valores que se mantienen constantes a lo largo del programa para determinadas variables basic destinadas a rótulos, sonidos musicales, colores, posicionamiento en pantalla, etc.

Las instrucciones entre la 100 y la 240 están destinadas a visualizar el rótulo de presentación deslizándolo de abajo hacia arriba dividiendo cada línea en dos partes, cada una de un color, y produciendo simultáneamente una música aleatoria restringida a 7 notas.

Una vez valorada la posición que deja el jugador, se emite un soniquete distinto en función de que la jugada realizada por el jugador sea buena y capaz de conducirle a la victoria (líneas de programa 1100-1140). El sonido será otro más corto si no se produce la condición necesaria pero no suficiente señalada en el punto anterior (líneas 910-850)

Al finalizar el juego se repiten las melodías a las que se refiere el párrafo anterior interpretando una u otra en función del resultado de la partida. Podría y debería haberse empleado para este fin sendas subrutinas, pero la sencillez de las instrucciones hacía más fácil su duplicación.
Para seguir el programa es aconsejable tener a la vista el siguiente cuadro de variables:

SONIDO

FO	Forma de onda
AF(I)	Alta frecuencia nota I
BF(I)	Baja frecuencia nota I

ROTULO

L$(I)	Contenido línea I del rótulo. Hay nueve líneas.

POSICIONAMIENTO

HT$	Horizontal
VT$	Vertical. Este se utiliza poco al ser más eficiente el empleo de TAB

COLORES

CL$	Gama de colores utilizados (6).
OC$	Color para la segunda parte de las dos en que se divide cada línea del rótulo de presentación.

JUEGO

BIT	Orden del bit examinado (0,1,2,3)
ESTADO	Valor del bit examinado (0,1)
SJ	Interruptor de juego. Toma el valor 1 si juega la máquina, 0 si corresponde jugar al operador.

COMUNICACION CON OPERADOR

RE$	Respuestas del operador en sentencias INPUT.
K$	Tecla pulsada por operador cuando está ejecutándose una sentencia GET.

CONTADORES DE BUCLES, MONTONES, POSICIONES, ETC.

I, J, K, L, M, sin que tengan una utilización constante a lo largo del programa.
El número de posiciones iniciales distintas es de $2 \cdot 16^4 = 131.072$.

```
10 DEF FN NB(X)=((2↑XANDF(1)) +(2↑XA
NDF(2)) + (2↑XANDF(3)) + (2↑XANDF(4)
))/(2↑X)
20 DEF FN PB(X)= FN NB(X)-2*INT((FN
NB(X))/2)
30 DEF FN VJ(X)= FN PB(0) OR FN PB(1
) OR FN PB(2) OR FN PB(3)
50 VT$="[40CRSRR]"
60 HT$="[HOM][25CRSRD]"
70 CL$="[BLK][WHT][PUR][GRN][YEL]":T
I$="000000":GOSUB 10100
100 PRINT"[CLR]":FOR I=25 TO -9 STEP
 -1
110 FOR J=0TO10
120 IF((J+I)<1)OR((J+I)>24)THEN 230
130 PRINTLEFT$(HT$,J+I);
140 K=INT((1+LEN(L$(J)))*RND(0))
150 PRINTMID$(CL$,RND(0)*5+1,1);
160 OC$= MID$(CL$,RND(0)*5+1,1)
170 PRINTLEFT$(L$(J),K) OC$MID$(L$(J
),K+1,40)
180 N=INT(1+7*RND(0))
190 POKE 54276,FO
200 POKE 54273,AF(N):POKE 54272,BF(N
)
210 POKE 54276,FO-1
220 PRINT LEFT$(HT$,24)"[RVSON]"STR$
(48-VAL(TI$))+"[SPC][RVSOFF]";
230 NEXT J,I
240 PRINT"[BLK]"
400 FOR J=1 TO 4:F(J)=1+INT(15*RND(0
)):NEXTJ
500 PRINT"[CLR][CRSRD]EL[SPC]JUEGO[SPC]
LO[SPC]GANA[SPC]QUIEN[SPC]QUITA[SPC]
LA[SPC]ULTIMA[2SPC]FICHA."
510 PRINT"[CRSRD]PUEDEN[SPC]QUITARSE
[SPC]UNA[SPC]VARIAS[SPC]O[SPC]TODAS[SPC]
LAS[2SPC]FICHAS[SPC]DE[SPC]UN[SPC]MO
NTON."
520 PRINT"[CRSRD]DEBE[SPC]ELEGIR[SPC]
TURNO[SPC]DE[SPC]JUEGO[SPC]PARA[SPC]
EMPEZAR[SPC]UD[SPC]O[SPC]LA[SPC]MAQU
INA."
530 PRINT"[CRSRD]LOS[SPC]MONTONES[SPC]
SE[SPC]GENERAN[SPC]ALEATORIAMENTE."
540 PRINT"[CRSRD]PARA[SPC]JUGAR[2SPC]
DEBE[SPC]INDICAR[2SPC]EL[SPC]NUMERO[2SPC]
DE[SPC]MONTON[SPC]DEL[SPC]QUE[SPC]QU
IERE[SPC]RETI";
550 PRINT"RAR[2SPC]FICHAS[SPC]Y[SPC]
EL[SPC]NUMERO[SPC]DE[SPC]ESTAS."
560 PRINT"[CRSRD]QUE[SPC]TENGA[SPC]S
UERTE[SPC]Y[SPC]QUE[SPC]APRENDA[SPC]
PRONTO[SPC]A[SPC]SER[SPC]EL[SPC]GANA
DOR."
570 PRINT"[4CRSRD][WHT]PARA[SPC]CONT
INUAR[SPC]PULSAR[SPC]UNA[SPC]TECLA"
580 GET K$:IF K$=""THEN 580
590 PRINT"[CLR][WHT]"
600 PRINTLEFT$(HT$,23)"[BLK]QUIEN[SPC]
JUEGA[SPC]PRIMERO[SPC]MAQUINA/HUMANO
":INPUT RE$
610 IF RE$="HUMANO"THEN SJ=0:GOTO 70
0
620 IF RE$="MAQUINA"THEN SJ=1:GOTO 7
00
630 PRINTLEFT$(HT$,23)"RESPUESTA[SPC]
NO[SPC]VALIDA:[SPC]MAQUINA/HUMANO"
640 FORI=1TO1000:NEXTI:GOTO600
700 PRINT "[CLR][BLK][CRSRD]"LEFT$(V
T$,16)"[RVSON][SPC]*[SPC]NIM[SPC]*[SPC]
[RVSOFF]"LEFT$(HT$,20);
710 PRINT "[WHT]"TAB(3)"[3COMMI]"TAB
(13)"[3COMMI]"TAB(23)"[3COMMI]"TAB(3
3)"[3COMMI]"
720 PRINT"[CRSRD][BLK]"TAB(3)F(1) TA
B(13)F(2) TAB(23)F(3) TAB(33)F(4)
730 FOR J=1 TO 4:IF F(J)=0 THEN 770
740 FOR I=1 TO F(J)
750 PRINT LEFT$(HT$,20-I)SPC(4+10*(J
-1))LEFT$(CL$,J)"[SHIFTQ]"
760 NEXT I
770 NEXT J
800 IF SJ=1THEN 900
810 PRINTLEFT$(HT$,23)"DE[SPC]DONDE[SPC]
QUITAS[SPC]FICHAS[SPC](1,2,3,4)[2SPC]
":INPUT RE$
820 IF LEFT$(RE$,1)<"1"OR LEFT$(RE$,
1)>"4"THEN 810
830 J=VAL(RE$):IF F(J)<1THEN PRINTLE
FT$(HT$,23)"NO[SPC]HAY[SPC]FICHAS[10SPC]
":GOTO896
840 PRINTLEFT$(HT$,23)"CUANTAS[2SPC]
QUITAS[19SPC]":INPUT RE$
850 IFRE$<"1"THENPRINTLEFT$(HT$,23)"
HAY[SPC]QUE[SPC]QUITAR[SPC]AL[SPC]ME
NOS[SPC]UNA[7SPC]":GOTO 895
860 IF VAL(RE$)>F(J)THENPRINTLEFT$(H
T$,23)"NO[SPC]HAY[SPC]TANTAS[SPC]EN[SPC]
MONTON" J:GOTO 895
870 F(J)=F(J)-VAL(RE$)
880 IF (F(1) OR F(2)OR  F(3) OR F(4)
) = 0 THEN 1300
890 SJ=1-SJ:GOTO 700
895 FOR K=1TO 500:NEXT K:GOTO 840
896 FOR K=1TO 500:NEXT K:GOTO 810
900 IF FN VJ(0)=0 THEN 1100
910 BIT=3:FORJ=1TO2:FORI=1TO2
920 POKE 54276,FO
930 POKE 54273,AF(I):POKE 54272,BF(I
)
940 POKE 54276,FO-1
950 NEXTI:FORI=1TO256:NEXTI,J
960 IF FN PB(BIT)=0THEN BIT=BIT-1:GO
TO960
970 J=1
```

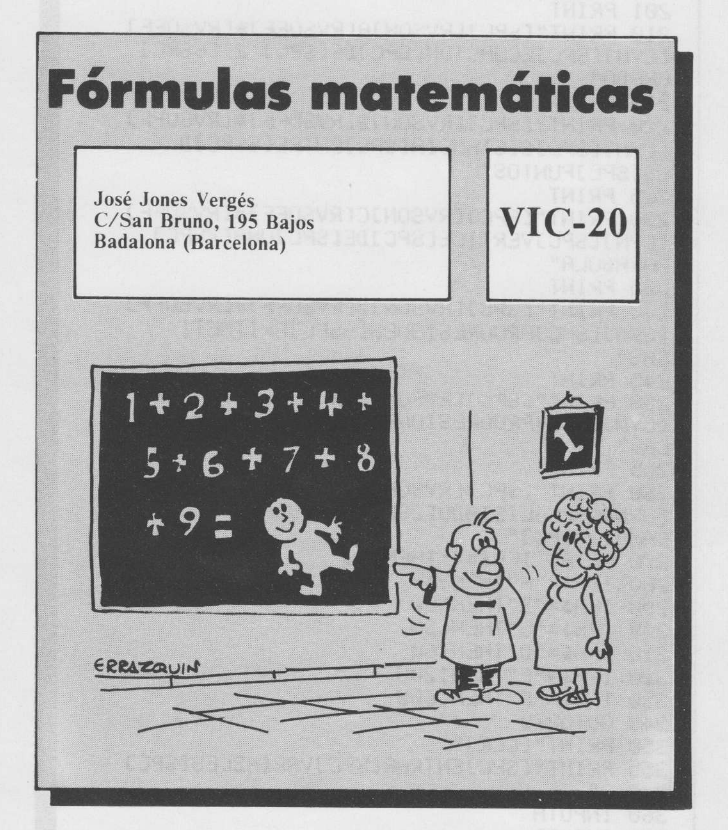
```
980 IF(F(J)AND2↑BIT)=0 THEN J=J+1:GO
TO980
990 F(J)=(F(J)AND(15-2↑BIT))
1000 IF BIT <=0 THEN 880
1010 BIT=BIT-1:IF FN PB(BIT)=0THEN10
00
1020 ESTADO=F(J)AND(2↑BIT):IF ESTADO
=0 THEN F(J)=(F(J)OR(2↑BIT)): GOTO 1
000
1030 GOTO 990
1100 FORK=0TO1:FOR L=1TO5:J=ABS(L-K*
5)+1
1110 POKE 54276,FO
1120 POKE 54273,AF(J):POKE 54272,BF(
J)
1130 POKE 54276,FO-1:FORI=1TO 32
1140 NEXT I,L,K
1200 J=INT(1+4*RND(0)):IF F(J)<1THEN
1100
1210 F(J)=INT(RND(0)*F(J)):GOTO 880
1300 IF SJ=0 THEN 1380
1310 PRINT HT$"JUEGO[SPC]Y[SPC]GANO"

1320 FORJ=1TO5:FORI=1TO2
1330 POKE 54276,FO
1340 POKE 54273,AF(I):POKE 54272,BF(
I)
1350 POKE 54276,FO-1
1360 NEXTI:FORI=1TO256:NEXTI,J
1370 GOTO 1440
1380 PRINT HT$"NO[SPC]ESTA[SPC]MAL[SPC]
...[2SPC]HAS[2SPC]GANADO[SPC]"
1390 FOR M=1 TO 4:FORK=0TO1:FOR L=1T
O5:J=ABS(K*5-L)+1
1400 POKE 54276,FO
1410 POKE 54273,AF(J):POKE 54272,BF(
J)
1420 POKE 54276,FO-1:FORI=1TO32
1430 NEXT I,L,K,M
1440 INPUT"DESEA[SPC]JUGAR[SPC]OTRA[SPC]
VEZ[SPC]S/N";RE$
1450 IF RE$="S"THEN 400
1460 IF RE$="N"THEN END
1470 GOTO 1440
9999 STOP
10100 L$(0)="[4SPC][2SHIFTQ][7SPC][2SHIFTQ]
[4SPC][2SHIFTQ][4SPC][2SHIFTQ][6SPC]
[2SHIFTQ]"
10120 L$(1)="[4SPC][3SHIFTQ][6SPC][2SHIFTQ]
[4SPC][2SHIFTQ][4SPC][3SHIFTQ][4SPC]
[3SHIFTQ]"
10140 L$(2)="[4SPC][4SHIFTQ][5SPC][2SHIFTQ]
[4SPC][2SHIFTQ][4SPC][4SHIFTQ][2SPC]
[4SHIFTQ]"
10160 L$(3)="[4SPC][2SHIFTQ][SPC][2SHIFTQ]
[4SPC][2SHIFTQ][4SPC][2SHIFTQ][4SPC]
[2SHIFTQ][SPC][4SHIFTQ][SPC][2SHIFTQ]
"
10180 L$(4)="[4SPC][2SHIFTQ][2SPC][2SHIFTQ]
[3SPC][2SHIFTQ][4SPC][2SHIFTQ][4SPC]
[2SHIFTQ][2SPC][2SHIFTQ][2SPC][2SHIFTQ]
"
10200 L$(5)="[4SPC][2SHIFTQ][3SPC][2SHIFTQ]
[2SPC][2SHIFTQ][4SPC][2SHIFTQ][4SPC]
[2SHIFTQ][6SPC][2SHIFTQ]"
10220 L$(6)="[4SPC][2SHIFTQ][5SPC][4SHIFTQ]
[4SPC][2SHIFTQ][4SPC][2SHIFTQ][6SPC]
[2SHIFTQ]"
10240 L$(7)="[4SPC][2SHIFTQ][6SPC][3SHIFTQ]
[4SPC][2SHIFTQ][4SPC][2SHIFTQ][6SPC]
[2SHIFTQ]"
10260 L$(8)="[4SPC][2SHIFTQ][7SPC][2SHIFTQ]
[4SPC][2SHIFTQ][4SPC][2SHIFTQ][6SPC]
[2SHIFTQ]"
10280 L$(9)="[35SPC]"
10300 FOR W=54272 TO 54296:POKEW,0:N
EXT
10310 FO=82:POKE 54296,15
10320 POKE 54277,9:POKE24278,240
10330 POKE 54275,0
10340 POKE 54274,255
10350 AF(1)=8:BF(1)=147
10360 AF(2)=10:BF(2)=205
10370 AF(3)=11:BF(3)=114
10380 AF(4)=12:BF(4)=216
10390 AF(5)=16:BF(5)=47
10400 AF(6)=14:BF(6)=10?
10410 AF(7)=8 :BF(7)=23
10420 RETURN
```

Fórmulas matemáticas

José Jones Vergés
C/San Bruno, 195 Bajos
Badalona (Barcelona)

VIC-20



Este programa que os envío es para el VIC-20; su aplicación esta hecha para las matemáticas de 1º y 2º de BUP. El programa da los títulos de algunas fórmulas matemáticas, apretando su tecla correspondiente, el programa pedirá las variables, que al ser dadas estas, el ordenador dará el resultado.

```
1 REM PROGRAMA FORMULAS MATEMATICAS
10 POKE36879,8
11 PRINT"[CLR]"
12 PRINT
13 PRINT
14 PRINT
15 PRINT
16 PRINT
20 PRINT"[4SPC][RVSON][RED][13COMM+]
"
30 PRINT"[4SPC][RVSON][RED][COMM+][SPC]
PROGRAMA[2SPC][COMM+]"
40 PRINT"[4SPC][RVSON][RED][COMM+][SPC]
REALIZADO[SPC][COMM+]"
50 PRINT"[4SPC][RVSON][RED][COMM+][SPC]
POR[SPC]....[2SPC][COMM+]"
60 PRINT"[4SPC][RVSON][RED][COMM+][SPC]
JOE[SPC]JONES[SPC][COMM+]"
70 PRINT"[4SPC][RVSON][RED][COMM+][4SPC]
(C)[4SPC][COMM+]"
```

```
80 PRINT"[4SPC][RVSON][RED][13COMM+]
"
90 FORT=1TO1980:NEXT:PRINT"[CLR]"
100 POKE36878,15
110 FORL=1TO100
120 POKE36876,INT(RND(1)*128)+128
130 FORM=1TO10
140 NEXTM
150 NEXTL
160 POKE36876,0
170 POKE36878,0
200 PRINT"[3SPC][RVSON][YEL]ESCOJER[SPC]
UNA[SPC]LETRA"
201 PRINT
210 PRINT"[SPC][RVSON]A[RVSOFF]*[RVSOFF]
[CYN][SPC]ECUACION[SPC]DE[SPC]'2'[6SPC]
GRADO"
215 PRINT
220 PRINT"[SPC][RVSON]B[RVSOFF]*[RVSOFF]
[CYN][SPC]DISTANCIA[SPC]ENTRE[6SPC]D
OS[SPC]PUNTOS"
225 PRINT
230 PRINT"[SPC][RVSON]C[RVSOFF]*[RVSOFF]
[CYN][SPC]VERTICE[SPC]DE[SPC]UNA[7SPC]
PARABOLA"
235 PRINT
240 PRINT"[SPC][RVSON]D[RVSOFF]*[RVSOFF]
[CYN][SPC]PROGRESIONES[9SPC]ARITMETI
CAS"
245 PRINT
250 PRINT"[SPC][RVSON]E[RVSOFF]*[RVSOFF]
[CYN][SPC]PROGRESIONES[9SPC]GEOMETRI
CAS"
255 PRINT
260 PRINT"[SPC][RVSON]F[RVSOFF]*[RVSOFF]
[CYN][SPC]LISTADO[2SPC]DEL[9SPC]PROG
RAMA[27SPC]"
270 GETA$:IFA$=""THEN270
280 IFA$="A"THEN350
290 IFA$="B"THEN500
300 IFA$="C"THEN630
310 IFA$="D"THEN760
320 IFA$="E"THEN1200
330 IFA$="F"THEN1600
340 GOTO270
350 PRINT"[CLR]"
355 PRINT"[SPC]ENTRAR[SPC]VARIABLES[SPC]
A,B,C"
360 INPUTA
370 INPUTB
380 INPUTC
390 S=SQR(B*B-4*A*C)
400 X= -B+S /2*A
410 Y= -B-S /2*A
420 PRINT"[SPC]LOS[SPC]RESULTADOS[SPC]
SON:"
423 PRINT
424 PRINT
430 PRINT;X
440 PRINT;Y
445 FORT=1TO9999:NEXT
450 PRINT"[CLR]"
460 GOTO200
500 PRINT"[CLR]"
510 PRINT"ENTRAR[SPC]VARIABLES[SPC]X
2,X1,Y2,Y1"
520 INPUTX2
530 INPUTX1
540 INPUTY2
550 INPUTY1
560 PRINT:PRINT
570 D=SQR((X2-X1)↑2+(Y2-Y1)↑2)
580 PRINT"[SPC]EL[SPC]RESULTADO[SPC]
ES:"
590 PRINTD
610 FORT=1TO9990:NEXT:PRINT"[CLR]"
620 GOTO200
630 PRINT"[CLR]"
640 PRINT"[SPC]ENTRAR[SPC]VARIABLES[SPC]
A,B,C"
```

```
650 INPUTA
660 INPUTB
670 INPUTC
680 PRINT:PRINT:PRINT
690 V1=-B/2*A
700 V2=4*A*C-B↑2/4*A
710 PRINT"[SPC]LOS[SPC]RESULTADOS[SPC]
SON:"
720 PRINTV1
730 PRINTV2
740 FORT=1TO9999:NEXT:PRINT"[CLR]"
750 GOTO200
760 PRINT"[CLR]"
770 PRINT:PRINT:PRINT:PRINT
780 PRINT"[SPC]ESCOJER[SPC]FORMULA"
785 PRINT:PRINT:PRINT
790 PRINT"[2SPC][RVSON]1[RVSOFF]*[RVSOFF]
[CYN]HALLAR[SPC]'AN'"
795 PRINT
800 PRINT"[2SPC][RVSON]2[RVSOFF]*[RVSOFF]
[CYN]HALLAR[SPC]LA[SPC]SUMA"
805 PRINT
810 PRINT"[2SPC][RVSON]3[RVSOFF]*[RVSOFF]
[CYN][SPC]VOLVER[SPC]A[SPC]SITUA-[6SPC]
CION[SPC]INICIAL"
820 GETA$:IFA$=""THEN820
830 IFA$="1"THEN860
840 IFA$="2"THEN960
850 IFA$="3"THEN8550
860 PRINT"[CLR]"
870 PRINT"ENTRAR[SPC]VARIABLES[SPC]A
1,N,D"
880 INPUTA1
890 INPUTN
900 INPUTD
910 AN=A1+(N-1)*D
911 PRINT:PRINT:PRINT
920 PRINT"[2SPC]EL[SPC]RESULTADO[SPC]
ES:"
930 PRINTAN
940 FORT=1TO9999:NEXT:PRINT"[CLR]"
950 GOTO760
960 PRINT"[CLR]"
970 PRINT"ENTRAR[SPC]VARIABLES[SPC]A
1,AN,N"
980 INPUTA1
990 INPUTAN
1000 INPUTN
1010 S=(A1+AN/2)*N
1020 PRINT:PRINT:PRINT
1030 PRINT"EL[SPC]RESULTADO[SPC]ES:"
1040 PRINTS
1050 FORT=1TO9999:NEXT:PRINT"[CLR]"
1060 GOTO760
1200 PRINT"[CLR]"
1210 PRINT:PRINT:PRINT:PRINT
1220 PRINT"ESCOJER[SPC]FORMULA"
1230 PRINT:PRINT:PRINT
1240 PRINT"[2SPC][RVSON]4[RVSOFF]*[RVSOFF]
[CYN][SPC]HALLAR[SPC]'AN'"
1245 PRINT
1250 PRINT"[2SPC][RVSON]5[RVSOFF]*[RVSOFF]
[CYN][SPC]HALLAR[SPC]SUMA"
1255 PRINT
1260 PRINT"[2SPC][RVSON]6[RVSOFF]*[RVSOFF]
[CYN][SPC]VOLVER[SPC]A[SPC]SITUA-[6SPC]
CION[SPC]INICIAL"
1265 PRINT
1270 GETA$:IFA$=""THEN1270
1280 IFA$="4"THEN1305
1290 IFA$="5"THEN1400
1300 IFA$="6"THEN8555
1305 PRINT"[CLR]"
1310 PRINT"[SPC]ENTRAR[SPC]VARIABLES
[SPC]A1,R,N"
1320 INPUTA1
1330 INPUTR
1340 INPUTN
1350 AN=A1*R↑(N-1)
```

```
1355 PRINT:PRINT:PRINT
1360 PRINT"EL[SPC]RESULTADO[SPC]ES:"

1370 PRINTAN
1380 FORT=1TO9999:NEXT:PRINT"[CLR]"
1390 GOTO1230
1400 PRINT"[CLR]"
1410 PRINT"[SPC]ENTRAR[SPC]VARIBLES[SPC]
AN,R,A1"
1420 INPUTAN
1430 INPUTR
1440 INPUTA1
1450 S=(AN*R-A1)/R-1
1455 PRINT:PRINT:PRINT
1460 PRINT"EL[SPC]RESULTADO[SPC]ES:"

1470 PRINTS
1480 FORT=1TO9999:NEXT:PRINT"[CLR]"
1590 GOTO1230
1600 PRINT"[CLR]"
1620 LIST
1680 PRINT:PRINT:PRINT
8550 PRINT"[CLR]":GOTO200
8555 PRINT"[CLR]":GOTO200
```

Rubik's Cube64

Pablo Cambra Brown
Martínez Valls, 14. P3
Onteniente (Valencia)

C-64

Este programa lo escribí para simular los giros del cubo de Rubik en la pantalla del ordenador.

No os asustéis por el hecho de que el título está escrito en inglés, la lengua no interviene para nada ya que las instrucciones para las teclas que realizan el giro de cada sección del cubo viene explicado con gráficos con flechitas en la parte superior de la pantalla una vez creados los caracteres necesarios.

Nada más comenzar el programa aparece un mensaje que nos pide paciencia debido a que el ordenador está creando caracteres.

Posteriormente se apaga la pantalla durante algunos segundos tras lo cual se ilumina con el cubo en el centro dibujado en tres dimensiones en el centro.

Las caras se llenan de sus respectivos colores (los gráficos son multicolores). Los colores son: rojo, blanco, azul, verde, amarillo y violeta.

Ahora el programa está listo para funcionar, propiamente dicho.

Hay 14 teclas que comandan 14 giros distintos. Aparte de las 14 teclas anteriormente citadas, está la "S" que devuelve el cubo a su forma original, es decir, una cara de cada color.

El programa está basado sobre los ARRAYS que, como se puede observar son numerosos y sobre el hecho de que ningún "cubito" que forma el cubo se mueve, sólo cambian de color. A pesar de que el programa hace uso de abundantes trucos para acelerar el BASIC, su funcionamiento sigue siendo lento, pero no lo suficiente para que sea tedioso, la "pega" es que hay un cierto desfase entre el apretón de la tecla y el giro realizado en el cubo.

```
2 CLR
5 DIMT(6,3,3),P(6,3,3),F(5),TT(4),XT
(3),YT(3)
10 GOTO280
20 FORG=1TO4
30 TT(G)=F(G+1)
40 NEXT
50 FORG=1TO4
60 FORD=1TO3:IFG=1THENIFKY>6THENIFKY
<13THENLM=4-SK:LYM=4-YT(D):L2=XT(D):
GOTO70
62 IFG=2THENIFKY>6THENIFKY<13THENL2=
4-SK:LYM=4-YT(D):LM=XT(D):GOTO70
65 LM=XT(D):LYM=YT(D):L2=XT(D)
70 T(TT(G),LM,LYM)=P(F(G),L2,YT(D))
80 NEXTD,G
90 IFAX=0THENRETURN
100 FORB=1TO3
110 FORC=1TO3
120 IFDI=0GOTO150
130 T(AX,C,B)=P(AX,4-B,C)
140 NEXTC,B:RETURN
150 T(AX,C,B)=P(AX,B,4-C)
160 NEXTC,B:RETURN
170 FORG=1TO4
180 TT(G)=F(G+1)
190 NEXTG
200 FORG=1TO3:IFKY=13GOTO230
205 T(TT(1),I1,G)=P(F(1),4-G,I1)
210 T(TT(2),G,I2)=P(F(2),I1,G)
215 T(TT(3),I2,G)=P(F(3),4-G,I2)
220 T(TT(4),G,I1)=P(F(4),I2,G)
225 NEXT:GOSUB100:RETURN
230 T(TT(1),G,I2)=P(F(1),I2,4-G)
235 T(TT(2),I1,G)=P(F(2),G,I2)
240 T(TT(3),G,I1)=P(F(3),I1,4-G)
245 T(TT(4),I2,G)=P(F(4),G,I1)
250 NEXT:GOSUB100:RETURN
280 PRINTCHR$(147):V=53248:POKEV+32,
7:POKEV+33,2
290 PRINTTAB(12)"[7CRSRD][BLU]***[COMM7]
PLEASE[SPC]WAIT[BLU]***"
295 PRINTTAB(8)"[2CRSRD]***[COMM7]CR
EATING[SPC]CHARACTERS[BLU]***"
300 FORG=0TO2000:NEXT
310 POKE53265,PEEK(53265)AND239
315 POKE53272,(PEEK(53272)AND240)+12

320 PRINTCHR$(142)
325 POKE52,48:POKE56,48
330 POKE56334,PEEK(56334)AND254
335 POKE1,PEEK(1)AND251
340 FORI=0TO1023:POKE12288+I,PEEK(V+
I):NEXT
380 POKE1,PEEK(1)OR4
385 POKE56334,PEEK(56334)OR1
387 FORI=12936TO12991:READCN:POKEI,C
N:NEXT
390 FORI=12992TO13111STEP2:READCN:PO
KEI,CN:POKEI+1,CN:NEXT
```

```
400 DATA0,126,126,126,126,126,126,0,
159,159,155,153,128,128,249,251
410 DATA249,249,217,153,1,1,159,223,
231,195,129,231,231,231,231,231
412 DATA231,231,231,231,231,129,195,
231,255,223,159,0,0,159,223,255
414 DATA255,251,249,0,0,249,251,255
420 DATA0,0,0,195,253,245,212,85,85,
127,127,127
430 DATA127,255,255,85,253,245,213,8
4,127,127,127,85
440 DATA85,253,253,253,253,253,253,8
5,84,64,0,0
450 DATA95,127,127,127 127,125,125,8
5,64,64,0,0,64,64,64,64
460 DATA0,0,0,85,64,64,0,0,9,10,14,1
5,12,13
470 FORI=1TO6
480 READCOL
490 FORA=1TO3:FORB=1TO3
500 T(I,A,B)=COL:NEXTB,A,I
510 POKEV+33,3:POKEV+35,0:POKEV+34,0

520 PRINTCHR$(147):CRM=V+2048
530 POKE53265,PEEK(53265)OR16
540 POKE53270,PEEK(53270)OR16
550 PRINT"[HOM][BLU][2SPC][SHIFTQ][SHIFTW]
[SHIFTQ][2SPC][3SHIFTQ][2SPC][3SHIFTQ]
[2SPC][SHIFTQ][SHIFTV][SHIFTQ][2SPC]
[3SHIFTQ][2SPC][3SHIFTQ][2SPC][3SHIFTQ]
"
560 PRINT".=[3SHIFTQ]/=[SHIFTQ][SHIFTW]
[SHIFTQ]0=[3SHIFTQ]1=[3SHIFTQ]2=[SHIFTQ]
[SHIFTV][SHIFTQ]3=[3SHIFTQ]4=[SHIFTT]
[2SHIFTQ]"
570 PRINT"[2SPC][3SHIFTQ][2SPC][3SHIFTQ]
[2SPC][SHIFTQ][SHIFTW][SHIFTQ][2SPC]
[3SHIFTQ][2SPC][3SHIFTQ][2SPC][SHIFTQ]
[SHIFTV][SHIFTQ][2SPC][3SHIFTQ]"
580 PRINT"[CRSRD][2SPC][3SHIFTQ][2SPC]
[3SHIFTQ][2SPC][3SHIFTQ][2SPC][3SHIFTQ]
[2SPC][3SHIFTQ][2SPC][3SHIFTQ][2SPC]
[3SHIFTQ]"
590 PRINT"5=[SHIFTQ][SHIFTT][SHIFTQ]
6=[2SHIFTQ][SHIFTT]7=[SHIFTU][2SHIFTQ]
8=[SHIFTQ][SHIFTU][SHIFTQ]9=[2SHIFTQ]
[SHIFTU]:=[SHIFTQ][SHIFTS][SHIFTQ];=
[SHIFTQ][SHIFTR][SHIFTQ]"
600 PRINT"[2SPC][3SHIFTQ][2SPC][3SHIFTQ]
[2SPC][3SHIFTQ][2SPC][3SHIFTQ][2SPC]
[3SHIFTQ][2SPC][3SHIFTQ][2SPC][3SHIFTQ]
"
990 PRINTTAB(18)"[COMM1][6COMMG]"
1000 PRINTTAB(9)"[BLU]LEFT[4SPC][COMM4]
[SHIFTY][SHIFT+][SHIFTY][SHIFT+][SHIFTY]
[SHIFT+][COMM-][COMM@][SPC][SHIFTZ]π
[SHIFTZ]π[SHIFTZ]π"
1010 PRINTTAB(9)"[BLU]SIDE[3SPC][COMM4]
[SHIFTY][SHIFT+][SHIFTY][SHIFT+][SHIFTY]
[SHIFT+][COMM-][COMMK][COMM@][SPC][SHIFT-]
[COMM*][SHIFT-][COMM*][SHIFT-][COMM*]
":REM←CHANGE SHIFT B
1020 PRINTTAB(9)"[BLU]VIEW[2SPC][COMM4]
[SHIFTY][SHIFT+][SHIFTY][SHIFT+][SHIFTY]
[SHIFT+][COMM-][COMMK][COMMI][COMM@]
[SPC][SHIFTZ]π[SHIFTZ]π[SHIFTZ]π"
1030 PRINTTAB(8)"[SHIFTZ]π[SHIFTZ]π[SHIFTZ]
π[SPC][SHIFTZ]π[SHIFTZ]π[SHIFTZ]π[COMMK]
[COMMI][COMMK][COMM@][SPC][SHIFT-][COMM*]
[SHIFT-][COMM*][SHIFT-][COMM*]"
1040 PRINTTAB(8)"[SHIFT-][COMM*][SHIFT-]
[COMM*][SHIFT-][COMM*][SPC][SHIFT-][COMM*]
[SHIFT-][COMM*][SHIFT-][COMM*][COMMI]
[COMMK][COMMI][COMM@][SPC][SHIFTZ]π[SHIFTZ]
π[SHIFTZ]π"
1050 PRINTTAB(8)"[SHIFTZ]π[SHIFTZ]π[SHIFTZ]
π[SPC][SHIFTZ]π[SHIFTZ]π[SHIFTZ]π[COMMK]
[COMMI][COMMK][COMM@][SPC][SHIFT-][COMM*]
[SHIFT-][COMM*][SHIFT-][COMM*]"
```

```
1060 PRINTTAB(8)"[SHIFT-][COMM*][SHIFT-]
[COMM*][SHIFT-][COMM*][SPC][SHIFT-][COMM*]
[SHIFT-][COMM*][SHIFT-][COMM*][COMMI]
[COMMK][COMMI][COMMT]"
1070 PRINTTAB(8)"[SHIFTZ]π[SHIFTZ]π[SHIFTZ]
π[SPC][SHIFTZ]π[SHIFTZ]π[SHIFTZ]π[COMMK]
[COMMI][COMM+][3SPC][BLU]REAR"
1080 PRINTTAB(8)"[SHIFT-][COMM*][SHIFT-]
[COMM*][SHIFT-][COMM*][SPC][SHIFT-][COMM*]
[SHIFT-][COMM*][SHIFT-][COMM*][COMMI]
[COMM+][4SPC][BLU]VIEW"
1085 PRINT
1090 FORI=0TO2:PRINTTAB(15)"[SHIFTZ]
π[SHIFTZ]π[SHIFTZ]π":PRINTTAB(15)"[SHIFT-]
[COMM*][SHIFT-][COMM*][SHIFT-][COMM*]
":NEXT
1100 PRINTTAB(3)"[4CRSRU][BLU]BOTTOM
[SPC]VIEW"
1110 FORSD=1TO6
1120 FORY=1TO3:FORX=1TO3:P(SD,X,Y)=T
(SD,X,Y):NEXTX,Y,SD
1150 GOTO1220
1170 FORG=PATOPBSTEPSA:Y=Y+IY
1180 FORX=2TO6STEP2
1190 POKECRM+G+(X-2),T(SD,X/2,INT(Y)
)
1200 POKECRM+G+(X-2)+1,T(SD,X/2,INT(
Y))
1210 NEXTX,G:RETURN
1220 SD=1:PA=338:PB=416:SA=39:IY=1:Y
=0:GOSUB1170
1230 SD=6:PA=448:PB=648:SA=40:IY=.5:
Y=.5:GOSUB1170
1240 SD=2:PA=455:PB=655:SA=40:IY=.5:
Y=.5:GOSUB1170
1250 SD=5:PA=346:PB=546:SA=40:IY=.5:
Y=.5:GOSUB1170
1260 SD=4:PA=735:PB=935:SA=40:IY=.5:
Y=.5:GOSUB1170
1270 PG=461
1280 FORX=1TO3:Y=0
1290 FORG=PGTOPG-78STEP-39:Y=Y+1
1300 POKECRM+G,T(3,Y,X)
1310 POKECRM+G+40,T(3,Y,X)
1320 NEXT
1330 PG=PG+80:NEXT:POKECRM+662,8
1350 GETA$:IFA$=""THEN1350
1360 IFA$<>"S"GOTO1400
1365 FORY=1TO3:FORX=1TO3:T(1,X,Y)=9
1367 P(1,X,Y)=9
1370 T(2,X,Y)=10:P(2,X,Y)=10
1372 T(3,X,Y)=14:P(3,X,Y)=14
1374 T(4,X,Y)=15:P(4,X,Y)=15
1376 T(5,X,Y)=12:P(5,X,Y)=12
1378 T(6,X,Y)=13:P(6,X,Y)=13
1380 NEXTX,Y:GOTO1220
1400 KY=ASC(A$)-45:REM PRINT"[BLK][HOM]
1400"
1410 IFKY<4THENIFKY>0GOTO1500
1420 IFKY<7THENIFKY>3GOTO1520
1430 IFKY<10THENIFKY>6GOTO1700
1440 IFKY<13THENIFKY>9GOTO1720
1450 IFKY=13THEN1900
1460 IFKY=14THEN1920
1470 GOTO1350
1500 F(1)=2:F(2)=3:F(3)=5:F(4)=6:F(5
)=2:REM-PRINT"[BLK][HOM]1500"
1510 GOTO1600
1520 F(1)=5:F(2)=3:F(3)=2:F(4)=6:F(5
)=5
1600 IFKY>3THENSK=KY-3:GOTO1620
1610 SK=KY
1620 FORG=1TO3
1630 YT(G)=SK:XT(G)=G:NEXT
1640 ONKYGOTO1650,1655,1660,1665,167
0
1645 AX=4:DI=1:GOSUB20:GOTO1110
1650 AX=1:DI=1:GOSUB20:GOTO1110
1655 AX=0:GOSUB20:GOTO1110
```

```
1660 AX=4:DI=0:GOSUB20:GOTO1110
1665 AX=1:DI=0:GOSUB20:GOTO1110
1670 AX=0:GOSUB20:GOTO1110
1700 F(1)=1:F(2)=5:F(3)=4:F(4)=2:F(5
)=1
1710 GOTO1730
1720 F(1)=4:F(2)=5:F(3)=1:F(4)=2:F(5
)=4
1730 K1=KY-6
1800 IFK1>3THENSK=K1-3:GOTO1820
1810 SK=K1
1820 FORG=1TO3
1830 XT(G)=SK:YT(G)=G:NEXT
1840 ONK1GOTO1850,1855,1860,1865,187
0
1845 AX=3:DI=1:GOSUB20:GOTO1110
1850 AX=6:DI=1:GOSUB20:GOTO1110
1855 AX=0:GOSUB20:GOTO1110
1860 AX=3:DI=0:GOSUB20:GOTO1110
1865 AX=6:DI=0:GOSUB20:GOTO1110
1870 AX=0:GOSUB20:GOTO1110
1900 F(1)=6:F(2)=1:F(3)=3:F(4)=4:F(5
)=6
1910 AX=2:I1=1:I2=3:DI=0:GOSUB170:GO
TO1110
1920 F(1)=1:F(2)=6:F(3)=4:F(4)=3:F(5
)=1
1930 AX=2:I1=3:I2=1:DI=1:GOSUB170:GO
TO1110
```



Este programa demuestra que las matemáticas nunca fallan y lo demuestra acertando la carta pensada después de varios pasos, el último paso es una pregunta que tendrás que responder con una cifra de dos o tres números, según la carta pensada.

Explicación del programa:

10 Variables para POKE
30- 90 Presentación del programa. Las líneas 60 y 80 gosub a subrutina de movimiento de la pantalla y efecto sonoro.
100-250 Mensajes para hacer las operaciones correspondientes de la carta pensada.
254-360 Representación de la carta.
370-600 Opinión para empezar otra vez el programa o finalizar.

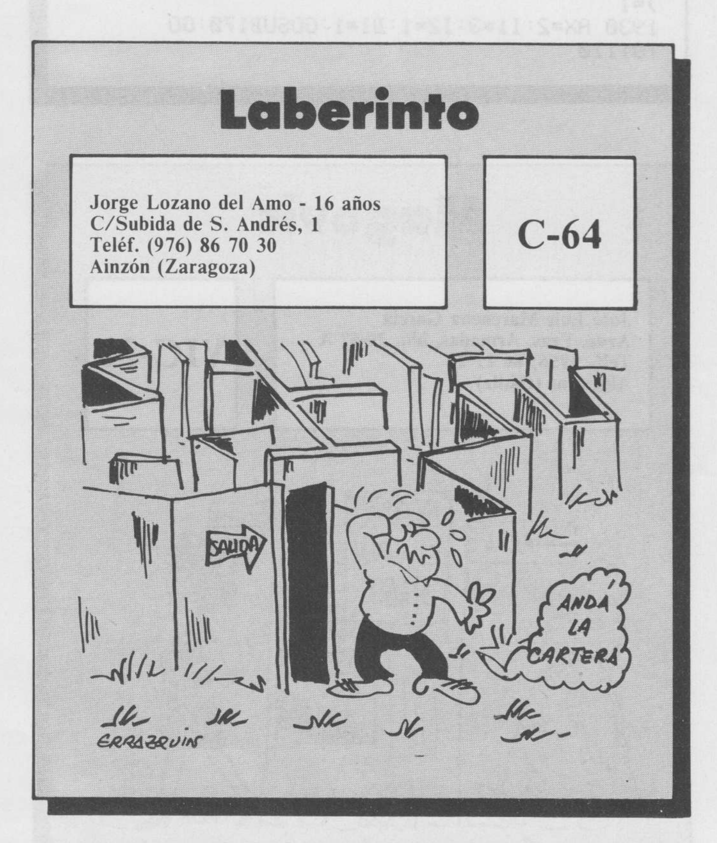
```
1 REM JUEGO MATEMATICO
2 REM J.L.MARCHENA
10 A=36878:B=36874:C=36879
30 POKEC,76:PRINT"[BLK]"
40 PRINT"[CLR]":POKE36865,140
50 PRINT"[10CRSRD][4SPC]M[SPC]A[SPC]
G[SPC]O[SPC]V[SPC]I[SPC]C[2SPC]"
60 GOSUB1999
61 POKE36875,200
62 FORT=1TO300:POKE36878,15:POKE3687
4,255:POKE36878,0:NEXT
65 PRINT"[CLR]":POKE36865,140
70 PRINT"[8CRSRD][9SPC]POR"
75 PRINT"[3CRSRD][SPC]JOSE[SPC]LUIS[SPC]
MARCHENA[SPC]G."
80 GOSUB1999
90 FORT=1TO300:POKEA,15:POKEB,128:PO
KEA,0:NEXT
100 PRINT"[CLR]":POKE36879,61
110 PRINT"[10CRSRD][3SPC]PIENSA[SPC]
UNA[SPC]CARTA":FORT=1TO20
114 POKEA,15:FORI=1TO75:POKEB,158:PO
KEA,0:NEXTI:NEXT
120 PRINT"[CLR][6CRSRD][SPC]MULTIPLI
CA[SPC]EL[SPC]NUMERO[SPC]"
130 PRINT"DE[SPC]DICHA[SPC]CARTA[SPC]
POR[SPC]DOS"
135 PRINT"[CRSRD][4SPC][RVSON][4SPC]
VALORES[4SPC]"
136 PRINT"[SPC][20COMM@]"
140 PRINT"[2CRSRD][BLU][SPC]AS=1;SOT
A=11;DAMA=12"
145 PRINT"[2CRSRD][8SPC]REY=13
150 FORT=1TO30:POKEA,15:FORI=1TO60:P
OKEB,158:POKEA,0:NEXTI:NEXT
160 PRINT"[CLR][6CRSRD][BLK][5SPC]SU
MALE[SPC]TRES[3SPC]"
165 FORT=1TO10:POKE36878,15:FORI=1TO
40:POKEB,158:POKEA,0:NEXTI:NEXT
170 PRINT"[5CRSRD]MULTIPLICALO[SPC]P
OR[SPC]CINCO"
175 FORT=1TO100:POKEA,15:FORI=1TO20:
POKEB,255:POKEA,0:NEXTI:NEXT
180 PRINT"[CLR][2CRSRD][SPC]SEGUN[SPC]
EL[SPC]PALO[SPC]DE[SPC]LA[SPC]"
190 PRINT"[CRSRD][SPC]CARTA,LE[SPC]S
UMARAS[SPC]:"
200 PRINT"[5CRSRD][BLU]1[BLK][CRSRR]
/SI[SPC]ES[SPC]DE[SPC]CORAZONES"
210 PRINT"[CRSRD][BLU]2[BLK][CRSRR]/
SI[SPC]ES[SPC]DE[SPC]ROMBOS"
220 PRINT"[CRSRD][BLU]3[BLK][CRSRR]/
SI[SPC]ES[SPC]DE[SPC]TREBOLES"
230 PRINT"[CRSRD][BLU]4[BLK][CRSRR]/
SI[SPC]ES[SPC]DE[SPC]PIQUES"
235 FORT=1TO200:POKEA,15:FORI=1TO10:
POKEB,158:POKEA,0:NEXTI:NEXT
240 PRINT"[CLR][4CRSRD][SPC]AHORA[SPC]
RESTALE[SPC]QUINCE"
245 PRINT"[2CRSRD]==================
===="
250 INPUT"[3CRSRD]QUE[SPC]NUMERO[SPC]
TE[SPC]DA";A$
254 PRINT"[CLR]":POKE36865,140:POKEC
,8:PRINT"[WHT]"
255 PRINT"[GRN][3SPC]LA[SPC]CARTA[SPC]
ES[SPC]EL[SPC]";
256 LETU=LEN(A$)
257 IFU=3THEN265
258 IFU=2THEN260
260 PRINTLEFT$(A$,1)
261 GOTO270
265 PRINTLEFT$(A$,2)
```

```
270 PRINT"[CRSRD][3SPC]DE[SPC]";
280 IFRIGHT$(A$,1)="1"THENPRINT"CORA
ZONES[12SPC]"
290 IFRIGHT$(A$,1)="2"THENPRINT"ROMB
OS"
300 IFRIGHT$(A$,1)="3"THENPRINT"TREB
OLES"
310 IFRIGHT$(A$,1)="4"THENPRINT"PIQU
ES"
320 PRINT"[4CRSRD]"
330 PRINT"[WHT][8SPC][COMMN][RVSON][3SPC]
[RVSOFF][COMMH]"
331 U=LEN(A$):IFU=2THEN334
332 IFU=3THEN336
334 Q$=LEFT$(A$,1)
335 GOTO340
336 Q$=LEFT$(A$,2)
340 IFQ$="10"THENQ$="0"
341 IFQ$="11"THENQ$="J"
342 IFQ$="12"THENQ$="Q"
343 IFQ$="13"THENQ$="K"
344 IFQ$="1"THENQ$="A"
349 PRINT"[8SPC][COMMN][RVSON]"Q$"[RVSON]
[2SPC][RVSOFF][COMMH]"
350 PRINT"[8SPC][COMMN][RVSON][3SPC]
[RVSOFF][COMMH]"
351 IFRIGHT$(A$,1)="1"THENPRINT"[8SPC]
[COMMN][RVSON][SPC][SHIFTS][SPC][RVSOFF]
[COMMH]"
352 IFRIGHT$(A$,1)="2"THENPRINT"[8SPC]
[COMMN][RVSON][SPC][SHIFTZ][SPC][RVSOFF]
[COMMH]"
353 IFRIGHT$(A$,1)="3"THENPRINT"[8SPC]
[COMMN][RVSON][SPC][SHIFTX][SPC][RVSOFF]
[COMMH]"
354 IFRIGHT$(A$,1)="4"THENPRINT"[8SPC]
[COMMN][RVSON][SPC][SHIFTA][SPC][RVSOFF]
[COMMH]"
355 PRINT"[8SPC][COMMN][RVSON][3SPC]
[RVSOFF][COMMH]"
356 PRINT"[8SPC][COMMN][RVSON][2SPC]
"Q$"[RVSOFF][COMMH]"
357 PRINT"[8SPC][COMMN][RVSON][3SPC]
[RVSOFF][COMMH]"
360 GOSUB1999
370 FORT=1TO100:POKEA,15:FORI=1TO5:P
OKEB,15:POKEA,0:NEXTI:NEXT
390 PRINT"[2CRSRD][GRN][SPC]QUIERES[SPC]
QUE[SPC]ADIVINE[SPC]"
400 PRINT"[CRSRD][8SPC]OTRA[SPC]CARTA
?";C$
410 GETA$:IFA$=""THEN410
420 IFA$="S"THENRUN
430 IFA$="N"THEN500
440 GOTO410
500 PRINT"[CLR]":POKE36865,140
505 PRINT"[6CRSRD]"
510 PRINT"[RVSON][22SPC][RVSOFF]";
520 PRINT"[RVSON][SPC][RVSOFF][3SPC]
[RVSON][SPC][RVSOFF][3SPC][RVSON][2SPC]
[RVSOFF][SPC][RVSON][SPC][RVSOFF][3SPC]
[RVSON][SPC][RVSOFF][3SPC][RVSON][3SPC]
[RVSOFF]";
530 PRINT"[RVSON][SPC][RVSOFF][SPC][RVSON]
[SPC][RVSOFF][SPC][RVSON][SPC][RVSOFF]
[SPC][RVSON][SPC][RVSOFF][2SPC][RVSON]
[SPC][RVSOFF][SPC][RVSON][SPC][RVSOFF]
[SPC][RVSON][SPC][RVSOFF][SPC][RVSON]
[SPC][RVSOFF][SPC][RVSON][5SPC][RVSOFF]
";
540 PRINT"[RVSON][SPC][RVSOFF][3SPC]
[RVSON][SPC][RVSOFF][SPC][RVSON][2SPC]
[RVSOFF][SPC][RVSON][SPC][RVSOFF][SPC]
[RVSON][SPC][RVSOFF][SPC][RVSON][SPC]
[RVSOFF][SPC][RVSON][SPC][RVSOFF][3SPC]
[RVSON][3SPC][RVSOFF]";
550 PRINT"[RVSON][SPC][RVSOFF][SPC][RVSON]
[SPC][RVSOFF][SPC][RVSON][SPC][RVSOFF]
[SPC][RVSON][2SPC][RVSOFF][SPC][RVSON]
[SPC][RVSOFF][SPC][RVSON][SPC][RVSOFF]
[SPC][RVSON][SPC][RVSOFF][SPC][RVSON]
[3SPC][RVSOFF][SPC][RVSON][3SPC][RVSOFF]
";
560 PRINT"[RVSON][SPC][RVSOFF][SPC][RVSON]
[SPC][RVSOFF][SPC][RVSON][SPC][RVSOFF]
[SPC][RVSON][SPC][RVSOFF][2SPC][RVSON]
[SPC][RVSOFF][SPC][RVSON][SPC][RVSOFF]
[SPC][RVSON][SPC][RVSOFF][SPC][RVSON]
[3SPC][RVSOFF][SPC][RVSON][4SPC][RVSOFF]
";
570 PRINT"[SPC][RVSON][SPC][RVSOFF][SPC]
[RVSON][SPC][RVSOFF][3SPC][RVSON][2SPC]
[RVSOFF][SPC][RVSON][SPC][RVSOFF][3SPC]
[RVSON][SPC][RVSOFF][3SPC][RVSON][SPC]
[RVSOFF][SPC][RVSON][SPC][RVSOFF]";
580 PRINT"[RVSON][22SPC][RVSOFF]"
600 GOSUB1999
1998 END
1999 POKE36878,15
2000 FORI=140TO40STEP-0.5:POKE36874,
I+115:POKE36865,I:NEXT
2100 POKE36878,0
2200 RETURN
```

Laberinto

Jorge Lozano del Amo - 16 años
C/Subida de S. Andrés, 1
Teléf. (976) 86 70 30
Ainzón (Zaragoza)

C-64



Os adjunto un programa llamado "LABERINTO", que yo he inventado.

Voy a comentaros las partes del programa:

0- 10 Inicio de variables, gosub para la presentación y nivel de dificultad.
15- 75 Dibuja aleatoriamente el laberinto y los componentes del juego, pone a cero el tiempo y pide un movimiento.
77-105 Asigna movimientos a las funciones, cuida que la pelota no se vaya de la pantalla y dibuja los movimientos.
132-155 Caso de que ganes.
190-240 Presentación del juego y posibles movimientos.
245-270 Choque con un muro.
290-310 Caso de que pierdas.
312-330 Opción para jugar de nuevo.

340-350 Si se alcanza un nuevo récord.
400-410 Sonido de los movimientos.
415-435 Aquí he puesto unas especificaciones para el listado, que se pueden suprimir.

Las vidas de que dispones para el juego van en relación con el nivel de dificultad, el doble de vidas del nivel escogido, las cuales están representadas en la parte superior derecha de la pantalla, y cada vez que se choca con un muro se pierde una de ellas.

El objeto del juego es llegar con tu pelota negra hasta el corazón rojo situado en el otro extremo, chocando las mínimas veces y en el menor tiempo posible.

```
0 D=60
1 REM*****************************
2 REM*                           *
3 REM*  LABERINTO: JORGE LOZANO  *
4 REM*                           *
5 REM*        JUNIO-1984         *
6 REM*                           *
7 REM*****************************
8 GOSUB200
9 INPUT"[2CRSRD][SPC]NIVEL[SPC]DE[SPC]
DIFICULTAD[SPC](1-9)";P:IFP<1ORP>9TH
EN8
10 V=54272:S=1064:Z=S+V:W=0:K=0:C=20
23:X=P*2:L=X:PRINT"[CLR]"
15 REM**** DIBUJO DE LA PANTALLA ***
*
20 PRINT"[HOM][BLU][RVSON]LABERINTO[SPC]
-[SPC]NIVEL:"P
25 FORLI=LTO1STEP-1:POKES-LI,90:POKE
Z-LI,6:NEXT:POKES-X-1,190
30 FORJ=1TO75*P
35 M=INT(RND(1)*958)
40 IFM=0THENM=1
45 IFPEEK(S+M)=160THEN35
50 POKES+M,160:POKEZ+M,5
55 NEXT
60 POKEC,83:POKEZ+959,2:TI$="000000"

65 IFPEEK(S+W)=160THENGOSUB250
70 POKES+W,81:POKEZ+W,0
75 GETA$:IFA$=""THEN75
77 REM**** ASIGNACION DE FUNCIONES *
***
80 IFA$="[F5]"THENPOKES+W,43:POKEZ+W
,4:W=W-1:K=K+1:GOSUB400
85 IFA$="[F6]"THENPOKES+W,43:POKEZ+W
,4:W=W+39:K=K+1:GOSUB400
90 IFA$="[F7]"THENPOKES+W,43:POKEZ+W
,4:W=W+1:K=K+1:GOSUB400
95 IFA$="[F8]"THENPOKES+W,43:POKEZ+W
,4:W=W+41:K=K+1:GOSUB400
100 IFA$="[F1]"THENPOKES+W,43:POKEZ+
W,4:W=W-40:K=K+1:GOSUB400
105 IFA$="[F2]"THENPOKES+W,43:POKEZ+
W,4:W=W-41:K=K+1:GOSUB400
110 IFA$="[F3]"THENPOKES+W,43:POKEZ+
W,4:W=W+40:K=K+1:GOSUB400
115 IFA$="[F4]"THENPOKES+W,43:POKEZ+
W,4:W=W-39:K=K+1:GOSUB400
117 IFA$="[HOM]"THEN1
120 IFC<S+WTHENW=W-40
125 IFC=S+WTHEN135
127 IFS>S+WTHENW=W+40
130 GOTO65
132 REM***** GANAS *****
135 PRINT"[CLR][RVSON][CRSRD][CRSRR]
T[CRSRD]U[CRSRR][CRSRU]G[CRSRD]A[CRSRD]
N[CRSRD]A[CRSRD]S[CRSRD]![CRSRD]![CRSRD]
!"
140 PRINT"[CRSRD][SPC]TIEMPO:[SPC]"M
ID$(TI$,4,1)"[SPC]MIN[SPC]"RIGHT$(TI
$,2)"[SPC]SEG"
145 IFD>TI/60THENGOSUB350
150 PRINT"[CRSRD][SPC]HAS[SPC]CHOCAD
O"X-L"VECES":PRINT"[CRSRD][SPC]MOVIM
IENTOS:";K
155 PRINT"[CRSRD][SPC]TIEMPO[SPC]REC
ORD:"D"SEG":GOSUB315
190 REM***** PRESENTACION *****
200 PRINT"[CLR]":POKE650,255:POKE532
80,5:POKE53281,1
202 PRINT"[CLR][2CRSRD][2SPC]LABERIN
TO"
205 PRINT"POR[SPC]J.LOZANO"
210 PRINT"[SPC](C)6-1984"
215 PRINT"[2CRSRD][11SPC]MOVIMIENTOS
:"
220 PRINT"[2CRSRD][9SPC]F2[4SPC]F1[4SPC]
F4"
225 PRINT"[2CRSRD][9SPC]F5[4SPC]**[4SPC]
F7"
230 PRINT"[2CRSRD][9SPC]F6[4SPC]F3[4SPC]
F8"
235 PRINT"[2CRSRD][SPC][HOME][SPC]VU
ELTA[SPC]A[SPC]EMPEZAR"
240 RETURN
245 REM**** CHOQUE CON UN MURO ****
250 FORZZ=1TO25:POKE53280,ZZ:POKE532
81,ZZ
255 POKEV+24,15:POKEV+4,17:POKEV+5,1
90:POKEV+6,248:POKEV+1,17:POKEV,37
260 POKEV+24,0:NEXTZZ:POKE53280,5:PO
KE53281,1
265 L=L-1:POKE1063-L,190:POKE1063+V-
L,0:IFL=0THEN300
270 RETURN
290 REM***** PIERDES *****
300 PRINT"[CLR][2CRSRD][RVSON][7CRSRR]
H[CRSRD][CRSRL]A[CRSRD][CRSRL]S[2CRSRD]
[CRSRL]P[CRSRD][CRSRL]E[CRSRD][CRSRL]
R[CRSRD][CRSRL]D[CRSRD][CRSRL]I[CRSRD]
[CRSRL]D[CRSRD][CRSRL]O[CRSRD][CRSRL]
![CRSRD][CRSRL]!"
305 PRINT"[CRSRD][SPC]HAS[SPC]CHOCAD
O"X"VECES"
310 PRINT"[CRSRD][SPC]EN[SPC]"MID$(T
I$,3,2)"[SPC]MIN[SPC]"RIGHT$(TI$,2)"
[SPC]SEG"
312 REM**** OPCION DE NUEVO JUEGO **
**
315 PRINT"[2CRSRD][SPC]JUEGAS[SPC]OT
RA[SPC](S/N)"
320 GETD$:IFD$="N"THENPRINT"[CLR]":E
ND
325 IFD$="S"THEN1
330 GOTO320
340 REM***** NUEVO RECORD *****
350 D=INT(TI/60):PRINT"[CRSRD][SPC]N
UEVO[SPC]TIEMPO[SPC]RECORD!![SPC]"RI
GHT$(TI$,2)"[SPC]SEG":RETURN
400 REM**** SONIDO ****
405 POKEV+24,15:POKEV+4,17:POKEV+5,1
90:POKEV+6,248:POKEV+1,17:POKEV,37
410 POKEV+24,0:RETURN
415 REM****** ESPECIFICACIONES *****
*
417 REM******     DEL LISTADO    *****
*
420 REM**"[CLR]"-ES [CLR]
421 REM**"[HOM]"-ES [HOME]
422 REM**"[CRSRD]"-ES [CRSRD]
423 REM**"[CRSRU]"-ES [CRSRU]
424 REM**"[CRSRR]"-ES [CRSRR]
425 REM**"[CRSRL]"-ES [CRSRL]
426 REM**"[RVSON]"-ES [RVSON]
427 REM**"[BLU]"-ES [BLUE]
428 REM**"[F1]"-ES F1
429 REM**"[F3]"-ES F3
430 REM**"[F5]"-ES F5
431 REM**"[F7]"-ES F7
432 REM**"[F2]"-ES F2
433 REM**"[F4]"-ES F4
434 REM**"[F6]"-ES F6
435 REM**"[F8]"-ES F8
```

Química

Paul Hernández K.
Plaza Gala Placidia 1-3, planta 15
Telf.: 237 63 54
Barcelona 6

VIC-20 + 3 K de expansión



El programa va de química ya que en él hay unos elementos químicos con los que se puede trabajar. Al poner en marcha el programa sale una pequeña presentación personal y luego el menú, desde ese punto ya todo es seguir las instrucciones del programa. Debido a causas ajenas a mi voluntad no he podido poner todos los elementos de la tabla periódica, pero hay 43 que creo que son bastantes. El programa usa ampliación (mínima 3 K) si no se posee ampliación suprimiendo algunos elementos creo que cabrá en la memoria estándar, el programa tal y como os lo mando yo ocupa exactamente 5.467 bytes (una vez cargados los datos).

```
1 REM PAUL HERNANDEZ K.
2 POKE36879,30:PRINT"[BLU]"
3 GOSUB5000
5 PRINT"[CLR]"
10 DIME$(43)
20 DIMN$(43)
30 DIMP$(43)
40 DIMV$(43)
50 FORT=1TO43
60 READE$(T),N$(T),P$(T),V$(T)
65 NEXTT
70 DATA HIDROGENO,H,1´008,1
80 DATA LITIO,L,6´94,1
90 DATA SODIO,NA,22´99,1
100 DATA POTASIO,K,39´10,1
110 DATA RUBUDIO,RB,85´47,1
120 DATA CESIO,CS,132´91,1
130 DATA COBRE,CU,63´54,1-2
140 DATA PLATA,AG,107´87,1
150 DATA ORO,AU,196´97,1-3
160 DATA BERILIO,BE,9´01,2
170 DATA MAGNESIO,MG,24´31,2
180 DATA CALCIO,CA,40´08,2
190 DATA ESTRONCIO,SR,87´62,2
200 DATA BARIO,BA,137´34,2
210 DATA RADIO,RA,226,2
220 DATA CINC,ZN,65´37,2
230 DATA CADMIO,CD,112´40,2
240 DATA MERCURIO,HG,200´59,1-2
250 DATA BORO,B,10´81,3
260 DATA ALUMINIO,AL,26´98,3
270 DATA CROMO,CR,51´99,2-3-6
280 DATA MANGANESO,MN,54´94,2-3-4-6-
7
290 DATA HIERRO,FE,55´85,2-3
300 DATA COBALTO,CO,58´93,2-3
310 DATA NIQUEL,NI,58´71,2-3
320 DATA CARBONO,C,12´01,4
330 DATA SILICIO,SI,28´09,4
340 DATA GERMANIO,GE,72´59,4
350 DATA ESTANO,SN,118´69,2-4
360 DATA PLOMO,PB,207´19,2-4
370 DATA NITROGENO,N,14´01,3-5
380 DATAFOSFORO,P,30´97,3-5
390 DATA ARSENICO,AS,74´92,3-5
400 DATA ANTIMONIO,SB,121´75,3-5
410 DATA BISMUTO,BI,208´98,3-5
420 DATA OXIGENO,O,15´99,2
430 DATA AZUFRE,S,32´06,4-6
440 DATA SELENIO,SE,78´96,4-6
450 DATA TELURO,TE,127´60,4-6
460 DATA FLUOR,F,18´99,1
470 DATA CLORO,CL,35´47,1-3-5-7
480 DATA BROMO,BR,79´91,1-3-5-7
490 DATA IODO,I,126´90,1-3-5-7
505 REM BUSQUEDAS
510 PRINT"[CLR][RVSON]***************
************[SPC]MENU[SPC]PRINCIPAL[SPC]
**************************"
515 PRINT"[2CRSRD][5SPC][RVSON]B[RVSOFF]
USQUEDAS"
517 PRINT"[CRSRD][5SPC][RVSON]L[RVSOFF]
ISTAR[SPC]ELEMENTOS"
519 PRINT"[CRSRD][5SPC][RVSON]T[RVSOFF]
EST"
521 PRINT"[CRSRD][5SPC][RVSON]F[RVSOFF]
IN"
523 PRINT"[CRSRD][5SPC][RVSON]V[RVSOFF]
ACIAR[SPC]MEMORIA"
525 GETO$:IFO$=""THEN525
527 IFO$="V"THEN20000
529 IFO$="B"THEN550
531 IFO$="F"THEN10000
533 IFO$="T"THEN15000
535 IFO$="L"THENGOTO1100
537 GOTO525
550 PRINT"[CLR][RVSON]****************
******************[SPC]MENU[SPC]******
**************************"
551 PRINT"[2CRSRD][3SPC]QUE[SPC]DESE
A[SPC]BUSCAR"
552 PRINT"[CRSRD][5SPC][RVSON]E[RVSOFF]
LEMENTOS"
553 PRINT"[CRSRD][5SPC][RVSON]N[RVSOFF]
OMENCLATURAS"
554 PRINT"[CRSRD][5SPC][RVSON]V[RVSOFF]
ALENCIAS"
555 GETO$:IFO$=""THEN555
556 IFO$="E"THEN600
557 IFO$="N"THEN700
558 IFO$="V"THEN800
559 GOTO555
560 GETO$:IFO$=""THEN560
600 PRINT"[CLR][RVSON]****************
*********[SPC]BUSQUEDA[SPC]ELEMENTOS
[SPC]**************************"
605 PRINT"[CRSRD]ELEMENTO[SPC]A[SPC]
BUSCAR"
610 INPUTA$
615 FORT=1TO43
620 IFA$=E$(T)THEN GOTO 650
630 NEXTT
640 PRINT"[CLR][CRSRD]EL[SPC]ELEMENT
O[SPC]";A$;
642 PRINT"[SPC]NO[SPC]EXISTE"
645 GOTO1000
650 PRINT"[CLR][CRSRD]ELEMENTO....:"
;A$
660 PRINT:PRINT"NOMENCLATURA:";N$(T)
```

```
670 PRINT:PRINT"PESO[SPC]ATOMICO:";P
$(T)
680 PRINT:PRINT"VALENCIAS... :";V$(T)

690 GOTO1000
700 PRINT"[CLR][RVSON]*************
********BUSQUEDA[SPC]NOMENCLATURAS**
*********************"
705 PRINT"[CRSRD]NOMENCLATURA[SPC]A[SPC]
BUSCAR"
710 INPUTA$
715 FORT=1TO43
720 IFA$=N$(T)THEN750
730 NEXTT
735 PRINT"[CLR][CRSRD]EL[SPC]ELEMENT
O[SPC]CUYA[SPC]"
740 PRINT"[CRSRD]NOMENCLATURA[SPC]ES
[SPC]";A$
745 PRINT"[CRSRD]NO[SPC]EXISTE"
747 GOTO1000
750 PRINT"[CLR][CRSRD]ELEMENTO....:"
;E$(T)
755 PRINT"[CRSRD]NOMENCLATURA:";A$
760 PRINT"[CRSRD]PESO[SPC]ATOMICO:";
P$(T)
770 PRINT"[CRSRD]VALENCIAS...:";V$(T
)
780 GOTO1000
800 PRINT"[CLR][RVSON]*************
*********[SPC]BUSQUEDA[SPC]VALENCIAS
[SPC]***********************"
805 PRINT"ENTRE[SPC]LA[SPC]COMBINACI
ON[SPC]"
807 PRINT"DE[SPC]VALENCIAS[SPC]A[SPC]
BUSCAR"
810 INPUTA$
812 PRINT"[CLR][3SPC]ELEMENTOS[SPC]C
ON[SPC]";A$
813 PRINT"[3SPC]VALENCIAS[CRSRD]"
815 FORT=1TO43
820 IFA$=V$(T)THENPRINT"[6SPC]";E$(T
)
830 NEXTT
840 GOTO1000
1000 PRINT"[3CRSRD][SPC]DESEA[SPC]MA
S[SPC]BUSQUEDAS"
1010 PRINT"[CRSRD][8SPC](S/N)"
1020 GETD$:IFD$=""THEN1020
1030 IFD$="S"THEN550
1035 IFD$="N"THEN15220
1040 GOTO1020
1100 PRINT"[CLR][RVSON]*************
*********";
1110 PRINT"[RVSON]*[SPC]LISTA[SPC]DE
[SPC]ELEMENTOS[SPC]*";
1120 PRINT"[RVSON]******************
****"
1130 PRINT"[7CRSRD][3SPC]PULSE[SPC]U
NA[SPC]TECLA[SPC]"
1140 GETH$:IFH$=""THEN1140
1150 FORT=1TO43
1160 PRINT"[CLR][CRSRD]ELEMENTO....
";E$(T)
1170 PRINT"[CRSRD]NOMENCLATURA:";N$(
T)
1180 PRINT"[CRSRD]PESO[SPC]ATOMICO:"
;P$(T)
1190 PRINT"[CRSRD]VALENCIAS...:";V$(
T)
1200 PRINT"[4CRSRD][3SPC]PULSE[SPC]U
NA[SPC]TECLA"
1210 GETI$:IFD$=""THEN1210
1220 NEXTT
1230 GOTO0505
5000 POKE37158,30:POKE37159,0
5005 PRINT"[CLR][RVSON]************
*********";
5010 PRINT"[RVSON]******[SPC]QUIMICA
[SPC]*******";
```

```
5020 PRINT"[RVSON]******************
****"
5030 PRINT"[CRSRD][SPC]BY:PAUL[SPC]H
ERNANDEZ[SPC]K."
5040 PRINT"[6CRSRD][3SPC]PULSE[SPC]U
NA[SPC]TECLA"
5050 POKE37159,66
5055 GETR$:IFR$=""THEN5055
5060 RETURN
10000 PRINT"[CLR][10CRSRD][8SPC]ADIO
S[9CRSRD]":POKE36879,27:END
15000 PK=1:PA=0
15001 PRINT"[CLR][RVSON]***********
*****************[SPC]TEST[SPC]****
****************************"
15010 PRINT"[CRSRD]YO[SPC]TE[SPC]DAR
E[SPC]UN[SPC]ELEMENTO"
15020 PRINT"Y[SPC]TU[SPC]ME[SPC]HAS[SPC]
DE[SPC]DECIR[SPC]"
15030 PRINT"[CRSRD]SU[SPC]NOMENCLATU
RA[SPC]Y[SPC]SUS"
15040 PRINT"[CRSRD]VALENCIAS[SPC]SI[SPC]
A[SPC]LA[SPC]TER-"
15045 PRINT"CERA[SPC]NO[SPC]LO[SPC]H
AS[SPC]ACERTA-"
15047 PRINT"DO[SPC]PASA[SPC]A[SPC]OT
RA[SPC]PREGUN-"
15048 PRINT"TA."
15050 PRINT"[2CRSRD][4SPC]PULSE[SPC]
UNA[SPC]TECLA"
15060 GETR$:IFR$=""THEN15060
15065 C=1
15070 T=INT(RND(1)*43)
15080 IFT=0THENT=43
15085 IFC=4THENPRINT"[CLR][CRSRD]YA[SPC]
HAN[SPC]PASADO[SPC]TUS[SPC]TRES[CRSRD]
[SPC]INTENTOS":GOTO25000
15090 PRINT"[CLR][CRSRD]ELEMENTO...:
";E$(T)
15100 INPUT"[CRSRD]NOMENCLATURA";M$
15110 INPUT"[CRSRD]VALENCIAS";B$
15120 IFM$<>N$(T)ORB$<>V$(T)THENPRIN
T"[5CRSRD][6SPC]NO":C=C+1:GOTO15125
15122 GOTO15130
15125 IFM$<> N$(T)THENPRINT"[2CRSRD]
[SPC]ES[SPC]OTRA[SPC]NOMENCLATURA"
15127 IFB$<>V$(T)THENPRINT"[2CRSRD][SPC]
ES[SPC]OTRA[SPC]VALENCIA"
15128 FORW=1TO2500:NEXTW
15129 GOTO15085
15130 PRINT"[3CRSRD]BIEN,HAS[SPC]ACE
RTADO":PA=PA+1
15140 PRINT "[3CRSRD]MAS[SPC]PREGUNT
AS[SPC](S/N)"
15145 GETQ$:IFQ$=""THEN15145
15150 IFQ$="S"THENPK=PK+1:GOTO15065
15160 IFQ$="N"THEN15200
15170 GOTO15140
15200 PRINT"[CLR][CRSRD]HAS[SPC]HECH
O";PK;"PREGUNTAS"
15205 PRINT"[CRSRD]HAS[SPC]ACERTADO.
:";PA
15210 PRINT"[CRSRD]HAS[SPC]FALLADO.:
";PK-PA
15220 PRINT"[2CRSRD][SPC]DESEA[SPC]V
OLVER[SPC]AL[SPC]MENU"
15225 PRINT"[CRSRD][6SPC](S/N)"
15230 GETZ$:IFZ$=""THEN15230
15240 IFZ$="S"THEN505
15250 IFZ$="N"THEN10000
15260 GOTO15210
20000 PRINT"[CLR][RVSON]***********
**********[SPC]BORRAR[SPC]PROGRAMA
[SPC]*************************"
20010 PRINT"[4CRSRD][2SPC]ESTAS[SPC]
SEGURO[SPC](S/N)"
20020 GETF$:IFF$=""THEN20020
20030 IFF$="S"THENSYS64802
20040 IFF$="N"THENGOTO505
```

```
20050 GOTO20020
25000 FORK=1TO1500:NEXTK
25010 PRINT"[CLR]LA[SPC]RESPUESTA[SPC]
CORRECTA[SPC][CRSRD]ERA:"
25020 PRINT"[2CRSRD]ELEMENTO....:";E
$(T)
25030 PRINT"[CRSRD]NOMENCLATURA:";N$
(T)
25040 PRINT"[CRSRD]VALENCIAS...:";V$
(T)
25050 GOTO15140
```

Devora corazones

Vicens Miquel y Juli González (15 años)
C/Mig, 5
Telf.: (972) 46 14 83
Cassa de la Selva (Gerona)

C-64

"Devora corazones" debe comer vitaminas para seguir viviendo y debe evitar unas trampas, además de plantas venenosas.

Explicación del programa:

40 Borra la pantalla: Pone la pantalla negra y el recuadro gris.
50 El cursor se pone verde y da el valor 0 a varias variables.
60- 170 Hace la presentación del programa y da instrucciones.
180- 200 Espera la pulsación del espacio para empezar.
220 Pregunta el nivel de dificultad.
230 El borde de la pantalla se vuelve amarillo y pone el título en la pantalla además del número de esta.
250- 280 Elige la situación de los diferentes caracteres en pantalla.
290 Coloca la posición inicial del Hearts Eater.
300- 355 Espera las teclas de movimiento y da valores a X e Y para que se vaya moviendo.
360- 376 Mira el carácter que el Hearts Eater tiene delante y según este sigue adelante, suma puntos o se destruye.
380 El carácter es un corazón. Resta 1 a la variable de los corazones CO, suma 5 a la puntuación, hace un ruido y vuelve al 300. Si la puntuación es o excede de 400, el Hearts Eater aumenta en 1, teniendo así un tanque más.
385 Hace lo mismo que la 380 excepto en que se trata de una vitamina.
390 Si se han comido todos los corazones y todas las vitaminas, pasa a otra pantalla.
400 Dibuja el Hearts Eater, hace un ruido y lo vuelve siempre blanco.
410 Hace la velocidad y mediante una resta, esta va aumentando a medida que pasa el tiempo.
430 Indica que el Hearts Eater ha muerto.
440 Resta un valor a la variable del Hearts Eater.
450 Si era el último de los tres Hearts Eater finaliza.
480 Aumenta el número de trampas y plantas venenosas al pasar a otra pantalla.
490 Indica el fin del juego.
500 Da la puntuación.
501 Si los puntos conseguidos son mayores que los anteriores te indica que has batido el récord.
503 Te pregunta el nombre.
504- 509 Hace una presentación del nombre del ganador y de los puntos conseguidos.
510- 550 Pregunta si quieres volver a jugar.
600- 620 Hace el ruido del movimiento del Hearts Eater.
1000-1040 Hace el ruido de fin de pantalla.

```
10 REM * HEARTS EATER *
20 J=0
40 PRINT"[CLR]":POKE 53281,0:POKE 53
280,12
50 POKE(646),13:CV=0:L=0:P=0
60 PRINT"[CRSRD]";TAB(12)"**********
******"
70 PRINTTAB(12)"*[SPC]HEARTS[SPC]EAT
ER[SPC]*"
80 PRINTTAB(12)"****************"
90 PRINT"[CRSRD]DEBES[SPC]COMERTE[SPC]
TODOS[SPC]LOS[SPC]CORAZONES[SPC][SHIFTS]
"
95 PRINT"[CRSRD]Y[SPC]TODAS[SPC]LAS[SPC]
VITAMINAS[SPC][SHIFTA]"
100 PRINT"[CRSRD]EVITANDO[SPC]CAER[SPC]
EN[SPC]LAS[SPC]TRAMPAS[SPC][COMMB]"
105 PRINT"[CRSRD]CHOCAR[SPC]CON[SPC]
TUS[SPC]SANGRIENTAS[SPC]HUELLAS[SPC]
[SHIFTQ]"
110 PRINT"[CRSRD]Y[SPC]COMER[SPC]LAS
[SPC]PLANTAS[SPC]VENENOSAS[SPC][SHIFTX]
"
120 PRINT"[CRSRD][RVSON]POR[SPC]V.[SPC]
MIQUEL[SPC]Y[SPC]J.[SPC]GONZALEZ"
130 PRINT"[CRSRD][RVSON][WHT]SUERTE[SPC]
!!!"
140 PRINT"DIRIGE[SPC]CON:"
150 PRINT"[CRSRD]A=ABAJO[10SPC]D=ARR
IBA"
160 PRINT"L=IZQUIERDA[7SPC];=DERECHA
"
170 PRINT"[2CRSRD][RVSON]PULSA[SPC]E
SPACIO[SPC]PARA[SPC]EMPEZAR"
180 GET T$:IF T$=""THEN180
190 IF T$<>"[SPC]" THEN 180
200 IF T$="[SPC]" THEN 220
220 PRINT"[CLR][5CRSRD]":INPUT"NIVEL
[SPC]DE[SPC]DIFICULTAD(1-9)";F:F=F*1
5:A=1:HE=3:R=0
230 POKE 53280,7:PRINT"[CLR][10CRSRD]
[10CRSRR][RVSON]HEARTS[SPC]EATER[SPC]
";A
240 H=5:CO=10:FORD=1TO2000:NEXT
250 PRINT"[CLR]":FORN=1TO10+R:W=INT(
RND(1)*1000):POKE1024+W,127:POKE5529
6+W,3:NEXTN
260 FORI=1TO10+R:WS=INT(RND(1)*1000)
:POKE1024+WS,88:POKE55296+WS,5:NEXTI
270 FORK=1TOCO:WR=INT(RND(1)*1000):P
OKE1024+WR,83:POKE55296+WR,2:NEXTK
275 FORSP=1TOH:WT=INT(RND(1)*1000):P
OKE1024+WT,65:POKE55296+WT,7:NEXTSP
```

```
280 FORKI=1TO600:NEXT
290 T=F:X=15:Y=5
300 GET M$
310 IF M$=""THENGOTO351
320 IF M$="A" THEN E=1
330 IF M$="D" THEN E=2
340 IF M$=";" THEN E=3
350 IF M$="L" THEN E=4
351 IF E=0 THEN X=X-1
352 IF E=1 THEN Y=Y+1
353 IF E=2 THEN Y=Y-1
354 IF E=3 THEN X=X+1
355 IF E=4 THEN X=X-1
360 IF PEEK(1024+X+40*Y)=83THEN380
361 IFPEEK(1024+X+40*Y)=65THEN385
365 IF PEEK(1024+X+40*Y)=81 THEN 430
370 IF PEEK(1024+X+40*Y)=88THEN 430
371 IF X=-1OR X=40 OR Y=-1OR Y=25 TH
EN 430
375 IF PEEK(1024+X+40*Y)=127THEN 430
376 IFPEEK(1024+X+40*Y)=32 THEN 400
380 CO=CO-1:P=P+5:IF P>=400ANDCV=0TH
ENHE=HE+1:CV=CV+1:GOSUB1000:GOSUB600

384 GOTO 390
385 H=H-1:P=P+6:IF P>=200ANDCV=0THEN
HE=HE+1:CV=CV+1:GOSUB1000:GOSUB600
390 IF CO=0 AND H=0 THEN 480
400 POKE 1024+X+40*Y,81:GOSUB600:POK
E55296+X+40*Y,1
410 FORZ=1TOT:NEXT:T=T-0.6
420 GOTO 300
430 POKE 1024+X+40*Y,90:GOSUB600
440 HE=HE-1
450 IF HE=0 THEN 490
460 GOTO 250
470 FOR U=1 TO 200:NEXT
480 FORTY=1TO4000:NEXT:FORHJ=1TO3:GO
SUB1000:NEXTHJ:R=R+7:A=A+1:GOTO 230
490 PRINT"[HOM][10CRSRD][8CRSRR][RVSON]
JUEGO[SPC]TERNINADO[SPC][RVSOFF]":PR
INT
500 FOR D=1 TO 2000:NEXT:PRINT"[CLR]
[4CRSRD][5CRSRR]TU[SPC]PUNTUACION[SPC]
";P
501 IF P>J THEN J=P:PRINT"[2CRSRD][RVSON]
[CYN]HAS[SPC]BATIDO[SPC]EL[SPC]RECOR
D[RVSOFF]":FORDE=1TO3000:NEXT:GOTO50
3
502 GOTO 510
503 INPUT"[CLR][10CRSRD][3CRSRR]ESCR
IBE[SPC]TU[SPC]NOMBRE";H$
504 PRINT"[CLR][4CRSRD][4CRSRR][RVSON]
[RED]EL[SPC]NUEVO[SPC]CAMPEON[SPC]ES
[SPC]:"
505 PRINT"[4CRSRD][4CRSRR][RVSON][BLU]
";H$
506 PRINT"[2CRSRD]*****************
**********************"
507 PRINT"[2CRSRD][3CRSRR]EL[SPC]REC
ORD[SPC]EN[SPC]PUNTUACION[SPC]ES"
508 PRINT"[CRSRD][4CRSRR][RVSON][YEL]
";J;"PUNTOS[RVSOFF]"
509 FORHG=1 TO4:GOSUB1000:NEXTHG:FOR
GY=1TO3000:NEXT
510 PRINT"[3CRSRD][2CRSRR]QUIERES[SPC]
VOLVER[SPC]A[SPC]JUGAR(S/N)[COMM6]"
520 GET G$
530 IF G$<>"S"AND G$<>"N" THEN 520
540 IF G$="S" THEN 40
550 IF G$="N" THEN END
600 POKE54296,15:POKE 54273,45
610 POKE54272,198
620 POKE 54296,0:RETURN
1000 V=54272:POKE 54296,15
1010 POKEV+6,8:POKEV+5,31
1020 POKE V+1,180:POKEV+4,33
1030 FOR SN=1 TO 100:NEXT
1040 POKEV+4,0:RETURN
```

Ranita

José Luis Armenteros León
C/ Luis de Camoens, 4-5º
Tels. 261003 (LA CORUÑA)

VIC-20

Es una versión del Frogger en que hay que atravesar una autopista y un río (teclas, A/Z.), si se ha conseguido cruzar la carretera y esquivar los submarinos hay que meterse en una de las casillas de la parte superior de la pantalla. Si se consigue ocupar todas las casillas se aumenta la puntuación (el marcador está en la esquina superior izquierda de la pantalla) y se consigue una vida más (las vidas están en la esquina inferior izquierda).

```
5 V=3:POKE36879,15:PRINT"[CLR]":A=81
74
10 B=8120:C=8034:D=7948:E=8097:F=800
9:R=15
20 H=7789:I=7812:J=7877:K=7900:M=791
0:Q=0
25 POKE52,28:POKE56,28
30 FORW=7168TO7679:POKEW,PEEK(W+2560
0):NEXT
40 POKE36869,255
50 FORZ=7168TO7215:READY:POKEZ,Y:NEX
T
60 FORW=38488TO38509:POKEW,2:NEXT
65 FORW=38576TO38597:POKEW,4:NEXT
70 FORW=38840TO38861:POKEW,2:NEXT
75 FORW=38752TO38773:POKEW,7:NEXT
80 FORW=38708TO38729:POKEW,6:NEXT
85 FORW=38664TO38685:POKEW,5:NEXT
88 FORW=38796TO38817:POKEW,5:NEXT
90 FORW=38620TO38641:POKEW,7:NEXT:PO
KE7745,4
91 FORW=7680TO7723:POKEW,4:NEXT
92 FORW=8166TO8186:POKEW,4:NEXT
93 FORW=7922TO7943:POKEW,4:NEXT
94 FORW=7723TO8163STEP44:POKEW,4:NEX
T
95 FORW=7702TO8164STEP44:POKEW,4:NEX
T
96 FORW=7724TO7744STEP2:POKEW,4:NEXT

97 POKEA,32:A=8174:IFV=1THENPOKE8164
,4:POKE8165,4:POKE38884,1:POKE38885,
1
```

```
98 IFV=2THENPOKE8164,0:POKE8165,4:PO
KE38884,3:POKE38885,1
99 IFV=3THENPOKE8164,0:POKE8165,0:PO
KE38884,3:POKE38885,3
100 POKEA,0:GETC$
110 IFC$="A"THENIFPEEK(A-44)<>4ANDPE
EK(A-44)<>0THENPOKEA,32:A=A-44:Q=Q+1
0:O=0
120 IFC$="Z"THENPOKEA,32:A=A+44:Q=Q-
20
130 IFC$=","THENIFPEEK(A-1)<>4THENPO
KEA,32:A=A-1
140 IFC$="."THENIFPEEK(A+1)<>4THENPO
KEA,32:A=A+1
150 IFA>8141THENA=8174
160 IFA<7746THENPOKEA,0:A=8174:P=P+1
:IFP=10THEN3000
170 POKEB,1:POKEB+1,2:POKEC,1:POKEC+
1,2:POKED,1:POKED+1,2:POKEE,3:POKEF,
3
180 POKEH,5:POKEI,5:POKEJ,5:POKEK,5:
POKEM,5
190 IFA=BORA=B+1ORA=B+2ORA=CORA=C+1O
RA=C+2ORA=DORA=D+1ORA=D+2ORA=EORA=E-
1ORA=FTHEN2000
200 IFA=F-1ORA=HORA=H-1ORA=IORA=I+1O
RA=JORA=J-1ORA=KORA=K+1ORA=MORA=M+1T
HEN2000
210 PRINT"[HOM]"Q
220 IFB=8139THENPOKEB,32:POKEB+1,32:
B=8120
230 IFC=8052THENPOKEC,32:POKEC+1,32:
C=8032
240 IFD=7965THENPOKED,32:POKED+1,4:D
=7944
250 IFE=8077THENPOKEE,32:E=8097
260 IFF=7988THENPOKEF,32:F=8009
270 IFH=7767THENPOKEH,4:H=7789
280 IFI=7833THENPOKEI,32:I=7812
290 IFJ=7856THENPOKEJ,32:J=7877
300 IFK=7922THENPOKEK,4:K=7900
310 IFM=7921THENPOKEM,32:M=7900
320 POKE36874,1:POKE36874,5
600 B=B+1:C=C+1:D=D+1:E=E-1:F=F-1:H=
H-1:I=I+1:J=J-1:K=K+1:M=M+1
610 POKEB-1,32:POKEC-1,32:POKED-1,32
:POKEE+1,32:POKEF+1,32:POKEH+1,32:PO
KEI-1,32
620 POKEJ+1,32:POKEK-1,32:POKEM-1,32
700 IFA=8174THENO=O+1:IFO=20THENPOKE
A,32:A=A-44:O=0
900 POKE36878,0:POKE36878,15:POKE368
76,125:POKE36875,135:POKE36874,145
1000 GOTO100
1010 DATA129,66,60,126,219,255,102,6
0,31,52,100,68,127,127,48,48,128,192
,96,48,254,254
1020 DATA24,24,12,12,60,76,255,88,21
9,195,255,129,189,165,165,189,129,25
5
1030 DATA24,16,16,16,126,255,255,126
2000 V=V-1:POKEA,0
2010 POKE36879,47:POKE36874,150:FORW
=1TO255:POKE36878,W:NEXT:POKE36874,0
:POKE36879,R
2020 IFV>0THEN97
2030 PRINT"[YEL][HOM][11CRSRD][9CRSRR]
FIN":END
3000 POKE36878,15:FORW=7680TO8185:PO
KEW,32:NEXT
3010 FORL=148TO220STEP.7
3020 POKE36876,L
3030 NEXTL
3040 FORL=128TO200
3050 POKE36876,L
3060 NEXTL
3070 FORL=200TO128STEP-1
3080 POKE36876,L
3090 NEXTL
3100 POKE36878,0
3110 POKE36876,0
3120 IFV<3THENV=V+1:POKE36879,8
3130 R=INT(RND(1)*6)+9:POKE36879,R
3140 P=0:O=0:GOTO60
```

Dibujo/Rosacea

J.M. González Uriarte
C/Umbe, 5-5º c-dcha.
Tel. (94) 469 05 69
Guecho (VIZCAYA)

C-64

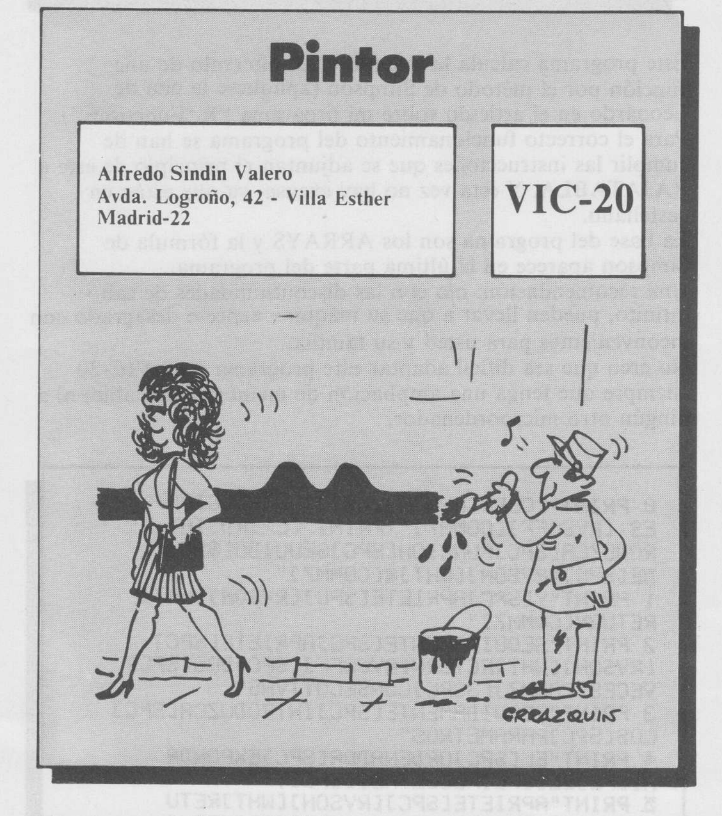
Dibuja una margarita con el radio que se desee. (Para valores mayores que 98 se sale de la pantalla), el número de puntos que se quiera (con más puntos, mayor nitidez en las líneas) y los pétalos que se nos antoje. (Según la norma dada en la presentación).

Comentarios del listado:

1- 11 Petición de datos.
12- 30 Protocolo de entrada en alta resolución.
40- 90 Paso a polares y fórmula de la rosácea.
100 Centrado en pantalla del origen de coordenadas.
110-170 Dibuja en alta resolución.
170-200 Fin.

```
1 PRINT"[CLR]":PRINTTAB(9);"DIBUJAND
O[SPC]LA[SPC]MARGARITA"
2 PRINTTAB(9);"[22COMMI]":PRINT:PRIN
T
3 PRINT"INDICAR[SPC]EL[SPC]RADIO(VAL
OR[SPC]MAX.98)";:INPUT A:PRINT:PRINT
:PRINT
4 PRINT"?CUANTOS[SPC]PUNTOS";:INPUT
N:PRINT:PRINT:PRINT
5 PRINT"PARA[SPC]DETERMINAR[SPC]EL[SPC]
NUMERO[SPC]DE[SPC]PETALOS[SPC]:":PRI
NT:PRINT
6 PRINT"*[SPC]SI[SPC]L[SPC]ES[SPC]IM
PAR,EL[SPC]NUM.[SPC]DE[SPC]PETALOS[SPC]
ES[SPC]L":PRINT
7 PRINT"*[SPC]SI[SPC]L[SPC]ES[SPC]PA
R,EL[SPC]NUM.[SPC]DE[SPC]PETALOS[SPC]
ES[SPC]2L":PRINT:PRINT:PRINT
8 PRINT"DAR[SPC]EL[SPC]VALOR[SPC]DE[SPC]
L";:INPUT L
```

```
9 PRINT"[CLR]":PRINT:PRINT:PRINT:PRI
NT:PRINT:PRINT
10 PRINTTAB(10);"ESPERA[SPC]UN[SPC]M
OMENTO..."
11 FOR W=0 TO 800:NEXT W :PRINT"[CLR]
"
12 BASE=2*4096:POKE53272,PEEK(53272)
OR 8
15 POKE 53265,PEEK(53265)OR 32
20 FOR I=BASE TO BASE+7999:POKEI,0:N
EXT
30 FOR I=1024 TO 2023:POKE I,3:NEXT
40 A2=2*π
60 IN=A2/N
70 FOR T=0 TO A2 STEP IN
80 R=A*COS(L*T)
90 X=R*COS(T):Y=R*SIN(T)
100 XP=160+X:YP=100-Y
110 RO=INT(YP/8)
120 CH=INT(XP/8)
130 LN=YP AND 7
140 BI=7-(XP AND 7)
150 BYP=BASE+RO*320+CH*8+LN
160 POKE BYP,PEEK(BYP)OR(2↑BI)
170 NEXTT
180 POKE 1024,16
190 GOTO190
200 END
```



Pintor

Alfredo Sindin Valero
Avda. Logroño, 42 - Villa Esther
Madrid-22

VIC-20

Este juego consiste en dirigir un hombrecillo mientras pinta toda la pantalla, escapando a la vez de un terrible monstruo. El juego tiene cinco niveles de dificultad, y cada vez que se completa una pantalla, el nivel es aumentado. Si llenamos tres pantallas en el nivel quinto, el ordenador se rinde (creo, porque aún no he podido llenar dos...). Por lo demás, el juego es autoexplicativo.

```
10 POKE36878,15:POKE36879,140:PRINTC
HR$(14)"[CLR][WHT]******[RVSON][SPC]
[SHIFTP][SHIFTI][SHIFTN][SHIFTT][SHIFTO]
[SHIFTR][SPC][RVSOFF]******"
11 PRINT"[SHIFTT]E[SPC]HAN[SPC]ENCAR
GADO[SPC]QUE[2SPC][CRSRD]PINTES[SPC]
DE[SPC]ROJO[SPC]EL[SPC]CAM-[CRSRD]PO
[SPC]DONDE[SPC]HABITA[SPC]EL[4SPC][CRSRD]
";
12 PRINT"*[SHIFTM][SHIFTO][SHIFTN][SHIFTS]
[SHIFTT][SHIFTR][SHIFTU][SHIFTO][SPC]
[SHIFTC][SHIFTO][SHIFTM][SHIFTE][SHIFTH]
[SHIFTO][SHIFTM][SHIFTB][SHIFTR][SHIFTE]
[SHIFTS]*[CRSRD][SHIFTP]ARA[SPC]MOVE
RTE[SPC]POR[SPC]LA[3SPC][CRSRD]ZONA,
TIENES[SPC]QUE[SPC]USAR[2SPC][CRSRD]
";
14 PRINT"EL[SPC]TECLADO:[11SPC][CRSRD]
[5CRSRR][RVSON]Z[RVSOFF]-[SHIFTA]BAJ
O."
15 PRINT"[CRSRD][5CRSRR][RVSON]X[RVSOFF]
-[SHIFTA]RRIBA.":PRINT"[CRSRD][5CRSRR]
[RVSON],[RVSOFF]-[SHIFTI]ZQDA.":PRIN
T"[CRSRD][5CRSRR][RVSON].[RVSOFF]-[SHIFTD]
CHA.[HOM]"
30 FORT=1TO15000:NEXT
40 PRINT"[CLR][SHIFTN]IVEL[SPC]DE[SPC]
JUEGO(1/5)[SPC]";:POKE204,0
42 GETA$:L=VAL(A$):IFL<1ORL>5THEN42
44 POKE204,1:ONLGOTO45,46,47,48,49
45 N=241:O=.8:S=0:GOTO50
46 N=241:O=.5:S=0:GOTO50
47 N=461:O=.5:S=0:GOTO50
48 N=461:O=.2:S=0:GOTO50
49 N=421:O=.25:S=1
50 DATA,,,,,,,,16,17,18,220,252,222,
17,28,28,17,223,252,223,17,28,
52 DATA28,28,8,255,29,29,36,66,56,56
,16,254,186,186,40,108,56,136,123,63
,59,72,136,8,56
54 DATA136,251,63,251,136,56,56,56,1
6,255,184,184,36,66,28,28,8,127,93,9
3,20,54,126
56 DATA126,126,126,126,126,126,126,
255,255,255,255,255,255,,36,153,90,6
0,189,102,60,198
58 DATA36,153,90,60,126,165,60,195,7
3,37,158,122,122,158,37,73,69,41,158
,122,122,158,41,69
60 RESTORE:FORI=7168TO7287:READQ:POK
EI,Q:NEXT:IFPEEK(56)<30THENPOKE56,28
:POKE52,28
62 FORI=7424TO7431:POKEI,0:NEXT:POKE
36869,255
80 PRINT"[HOM][RVSON][WHT][22COMM+][RVSOFF]
";:
82 IFS=0THEN:FORI=0TO20:PRINT"@@@@@@
@@@@@@@@@@@@@@@@";:NEXT
84 IFS=1THEN:FORI=0TO20:PRINT"[RVSON]
[COMM+][RVSOFF]@@@@@@@@@@@@@@@@@@@@[RVSON]
[COMM+][RVSOFF]";:NEXT
86 IFL>=3THENPRINT"[RVSON][21COMM+][HOM]
";
88 IFL<3THENPRINT"[HOM][12CRSRD][RVSON]
[22COMM+]";:FORI=1TO9:PRINT"[22COMM+]
";:NEXT
89 POKE7922,0:POKE7921,0
90 P=7922:C=30720:F=7935:POKEP,4:POK
EP+C,1:H=0:NC=0
100 A=PEEK(203)
110 A=INT((A-26)/3):D=22*(A=0)+(A=1)
-22*(A=2)-(A=3):E=PEEK(P+D):IFEANDE<
>9ANDE<>10THEN130
115 IFPEEK(P+D)=0THENNC=NC+1:POKE368
77,250:FORT=1TO10:NEXT:POKE36877,0:I
FNC>=NTHEN250
120 POKEP,9+(AAND1):POKEP+C,2:P=P+D:
POKEP,2*A+1:POKEP+C,1:POKEF,PEEK(F)+
1
130 IFPEEK(F+R)<13ANDRND(TI)<OTHEN18
0
140 B=-(INT((F-7680)/22)=INT((P-7680
)/22))
```

```
150 R=-B*(P>F)+B*(P<F)+22*(B=0ANDP<F
)-22*(B=0ANDP>F):IFDTHENPOKEP,2*A+2
160 IFPEEK(F+R)>13THENB=INT(RND(TI)*
4):R=(B=0)-(B=1)-22*(B=2)+22*(B=3):G
OTO160
170 IFP=F+RTHEN200
180 POKEF,H:POKEF+C,J:F=F+R:H=PEEK(F
):J=PEEK(F+C):POKEF,13-2*(RAND1):POK
EF+C,7
181 IFDTHENPOKEP,2*A+2
185 FORI=0TO3:POKE36876,250-30*I:NEX
T:POKE36876,0
190 IFH>0ANDH<9THENPOKEF,H:POKEF+C,J
:F=F-R:POKEF,13-2*(RAND1):POKEF+C,7:
GOTO200
195 GOTO100
200 CA=PEEK(P):FORI=0TO7:POKE7168+I+
8*CA,255:POKE36876,250-10*I
205 FORT=1TO50:NEXTT,I
210 POKE36876,0:PRINT"[HOM][RVSON]HA
[SPC]SIDO[SPC]CAPTURADO[SPC]!!![SPC]
";:POKE36874,150:FORT=1TO2000:NEXT:P
OKE36874,0
220 GOTO50
250 POKEP,9+(AAND1):POKEP+C,2:P=P+D:
POKEP,2*A+1:POKEP+C,1:PRINT"[HOM][RVSON]
BIEN[SPC]HECHO![11SPC]"
260 POKE36876,230:FORT=1TO1000:NEXT:
POKE36876,0:IFL<5THENPRINT"[HOM][RVSON]
AUMENTO[SPC]LA[SPC]DIFICULTAD[SPC]"
270 L=L-(L<5):IFL=5THENCO=CO+1:IFCO=
4THENPRINT"[HOM][RVSON][22COMM+]"
280 FORT=1TO2000:NEXT:IFCO<4THEN44
300 POKE36869,240:PRINTCHR$(14)"[CLR]
[SHIFTB][SHIFTR][SHIFTA][SHIFTV][SHIFTO]
,[SHIFTH][SHIFTA][SPC][SHIFTL][SHIFTO]
[SHIFTG][SHIFTR][SHIFTA][SHIFTD][SHIFTO]
[SPC][SHIFTG][SHIFTA][SHIFTN][SHIFTA]
[SHIFTR]"
310 POKE36869,240:PRINTCHR$(14)"[CLR]
[SHIFTB][SHIFTR][SHIFTA][SHIFTV][SHIFTO]
,[SHIFTH][SHIFTA][SPC][SHIFTL][SHIFTO]
[SHIFTG][SHIFTR][SHIFTA][SHIFTD][SHIFTO]
[SPC][SHIFTG][SHIFTA][SHIFTN][SHIFTA]
[SHIFTR]"
320 PRINT"[SHIFTE]SPERO[SPC]QUE[SPC]
SE[SPC]HAYA[SPC]DI-[CRSRD]VERTIDO.[SHIFTN]
OS[SPC]VEREMOS!!"
330 PRINT"[CRSRD][SHIFTQ]UIERE[SPC]J
UGAR[SPC]MAS(S/N)";:POKE204,0
340 GETA$:IFA$<>"S"ANDA$<>"N"THEN340

350 IFA$="S"THENRUN
```



Carga de Programas

Cuando queremos cargar un programa desde otro, la longitud del segundo ha de ser menor que la del primero, o no se cargará correctamente. Para solucionar esto hay que escribir en vez del "LOAD" lo siguiente:

POKE631,131:POKE198,1:END

Produce el mismo efecto que SHIFT/RUN STOP

G.S. RUN EEUU.

Errores de carga

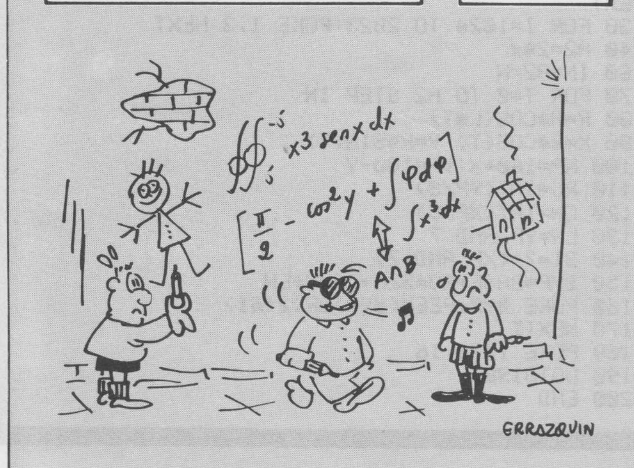
Si recibes demasiados errores en la carga de un programa, puede ser debido a que el cassette este demasiado cerca del televisor. Prueba a alejarlo.

R.D.B. RUN EEUU.

Simpson

Pablo Cambra Brown
Martínez Valls, 14. P3
Onteniente (Valencia)

C-64



Este programa calcula la integral en el intervalo de una función por el método de Simpson (aplíquese la cita de Leonardo en el artículo sobre mi programa "R. Functions").
Para el correcto funcionamiento del programa se han de cumplir las instrucciones que se adjuntan al principio de este a RAJATABLA. Y esta vez no hay excusa, ya que están en castellano.
La base del programa son los ARRAYS y la fórmula de Simpson aparece en la última parte del programa.
Una recomendación: ojo con las discontinuidades de salto infinito, pueden llevar a que su máquina exprese desagrado con inconvenientes para usted y su familia.
No creo que sea difícil adaptar este programa a un VIC-20 (siempre que tenga una ampliación de memoria aceptable) ni a ningún otro microordenador.

```
0 PRINT"[CLR][RVSON][WHT]INSTRUCCION
ES:[RVSOFF][COMM7]":PRINT"[CRSRD]INT
RODUZCA[SPC]FUNCION[SPC]SEGUIDO[SPC]
DE[SPC][RVSON][WHT]@[COMM7]"
1 PRINT"Y[SPC]APRIETE[SPC][RVSON][WHT]
RETURN[COMM7]"
2 PRINT"SEGUIDAMENTE[SPC]APRIETE[SPC]
[RVSON][WHT]RETURN[RVSOFF][SPC]DOS[SPC]
VECES[COMM7][3SPC]CONSECUTIVAS"
3 PRINT"SEGUIDAMENTE[SPC]INTRODUZCA[SPC]
LOS[SPC]PARAMETROS"
4 PRINT"EL[SPC]ORDENADOR[SPC]EXPONDR
A[SPC]EL[SPC]RESULTADO[SPC]Y"
5 PRINT"APRIETE[SPC][RVSON][WHT]RETU
RN[COMM7][RVSOFF][SPC]OTRA[SPC]VEZ."

6 PRINT"[2CRSRD][RVSON]APRIETE[SPC]U
N[SPC]TECLA"
10 GETO$:IFO$=""THEN10
15 PRINT"[CLR][9SPC][SHIFTU]B[2CRSRD]
[2CRSRL][SHIFTK]A[2CRSRU]":CLR
20 INPUT"INTEGRAL[CRSRR][SHIFTB][2CRSRL]
";A$
60 PRINT"[4CRSRD]162DEFFNSM(X)="A$"[CRSRL]
[SPC]":PRINT"RUN105":PRINT"[5CRSRU]"
:END
```

```
105 E=EXP(1)
110 DX=1
130 INPUT"PRIMER[SPC]PUNTO[SPC]DEL[SPC]
INTERVALO(A)";A
140 INPUT"SEGUNDO[SPC]PUNTO[SPC]DEL[SPC]
INTERVALO(B)";B:PRINT
150 INPUT"[CRSRU]NUMERO[SPC]DE[SPC]D
IVISIONES[7SPC][7CRSRL]";M
153 IFM>3750THENM=3750:PRINT"NUMERO[SPC]
DE[SPC]DIVISIONES[SPC]LIMITADO[SPC]A
:"M:GOTO162
155 IFM<=0OR(M-INT(M))<>0THEN150
170 DIMY(2*M)
175 X=A
180 IX=(B-A)/(2*M)
190 FORR=1TO2*M
200 X=X+IX
210 Y(R)=FNSM(X)
220 NEXT
240 S1=0
250 S2=0
260 FORR=1TO2*MSTEP2
270 S1=S1+Y(R)
280 NEXT
290 FORR=2TO2*MSTEP2
300 S2=S2+Y(R)
310 NEXT
315 X=A:Y0=FNSM(X)
320 X=B:Y2M=FNSM(X)
330 F=(IX/3)*(Y0+Y2M+2*S2+4*S1)
332 PRINT"[CRSRD]LA[SPC]SOLUCION[SPC]
ES:"
334 PRINT"[CRSRD]"B:PRINT"[CRSRD]"A
335 PRINT"[2CRSRU][SHIFTB][CRSRU][CRSRL]
[SHIFTU][2CRSRD][CRSRL][SHIFTK][CRSRU]
";
336 FORT=1074TO1100
337 IFPEEK(T)=0THEN400
338 IFPEEK(T)<32THENPRINTCHR$(PEEK(T
)+64);:GOTO350
340 PRINTCHR$(PEEK(T));
350 NEXT
400 PRINT"=";F
500 PRINT"[3CRSRD]162[3CRSRU]":END
```



Comando SAVE

Con el comando SAVE (grabar) podemos conseguir curiosos efectos; por ejemplo:

SAVE"[CLR] NOMBRE DEL PROGRAMA"

Cuando el ordenador encuentra este programa, la pantalla se borra y el nombre aparece en la parte superior. Efectos parecidos se producen con un [RVS ON], [CTRL+COLOR], etc. antes del nombre del programa.

A.I. RUN EE.UU.

Par o impar

Una manera fácil de saber si un número es par o impar es utilizando el operador lógico AND. Si el resultado es uno, el número es impar, de lo contrario es par.

Ej. PRINT 234 AND 1; nos responderá 0; 437 AND 1 nos responderá 1.

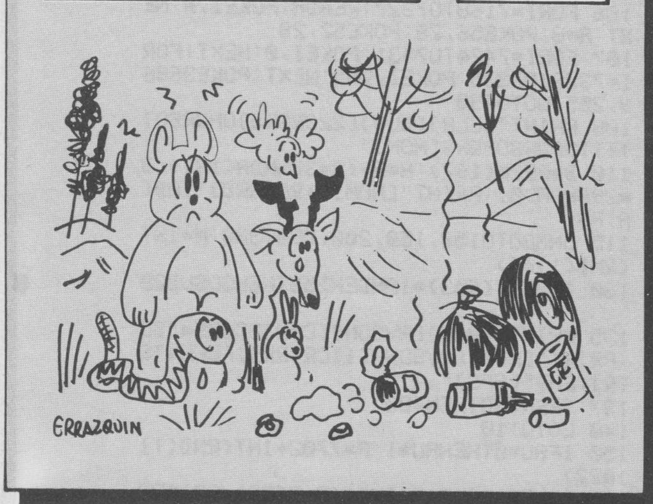
Este truco sólo sirve para números comprendidos entre -32768 y 36767, si nos salimos de estos límites el ordenador nos dará un ILLEGAL QUANTITY ERROR.

W.C.N. RUN EEUU.

Invasores

Alfredo Sindin Valero
Avda. Logroño, 42 - Villa Esther
Madrid-22

VIC-20



El título ya es bastante explicativo. Manejando una base láser, debemos destruir los distintos tipos de invasores que tratan de destruirnos y, de paso, invadir nuestro planeta.

Los mutantes se dirigen directamente a nuestra nave para absorber nuestra energía, pero saben apartarse de los láser...

Los cazas disparan rayos que, de alcanzarnos, nos quitarán la mitad de la energía disponible, y el único interés de los transportes es aterrizar en la explanada de nuestra zona defensiva. Destruir estos invasores nos dará puntos y energía.

Este programa funciona en el VIC sin expandir y, con ligerísimas modificaciones, puede funcionar en una máquina con 16 K. Dejo a la habilidad de los interesados el desarrollo de estas modificaciones.

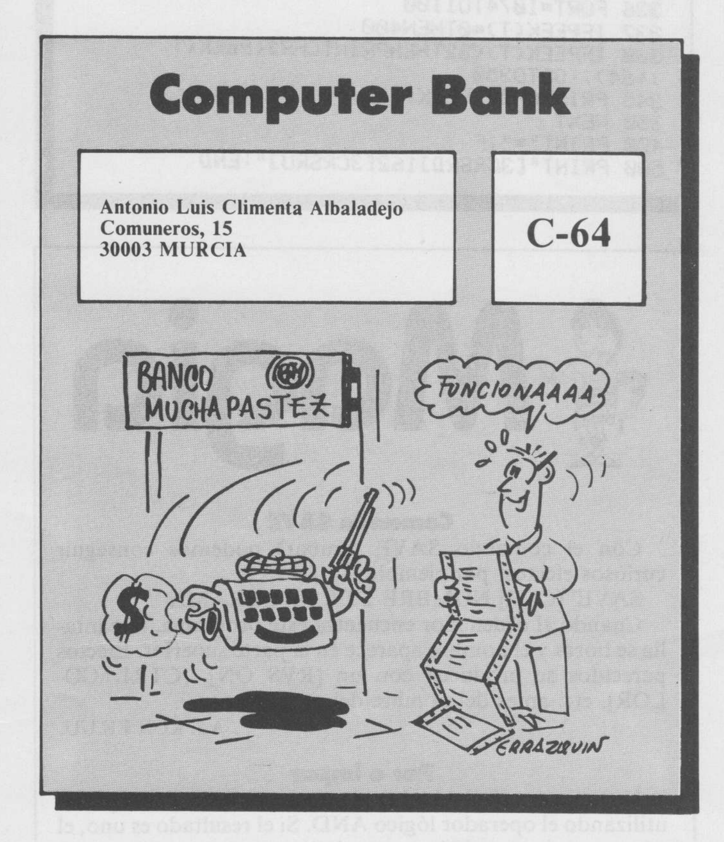
```
5 POKE36878,15:R=36877:R1=36876:CL=7
:EN=1000:S1=36875:A$="[SPC][YEL]EF[SPC]
[CRSRD][4CRSRL][SPC]GH[SPC]"
10 POKE36879,8:PRINT"[CLR]":C=30720:
GOTO100
20 EN=EN-50:FORI=8077+ATO7703+ASTEP-
22:F=I:S=S+5:IFPEEK(I)<5THENGOSUB45
22 IFI>7703+ATHENPOKEI,0:POKEI+C,2:P
OKER,250-S
23 NEXT
24 FORD=8077+ATOFSTEP-22:POKED,32:NE
XT:POKE36877,0:S=0:RETURN
45 W=PEEK(I):SC=SC-10*(W=4)-20*(W=3)
-30*(W=2)
50 FORE=9TO14:POKEI,E:POKER1,110+10*
E:POKEI+C,CL:CL=14/CL:FORTA=1TO20:NE
XTTA,E
55 POKEI,32:POKER1,0:I=7703+A:AU=0:M
=0:DE=1:EN=EN+100:RETURN
100 DATA2,1,2,1,2,1,2,1,16,8,16,8,16
,8,16,8,102,255,153,153,255,153,66,3
6,36,66,153,231
101 DATA231,153,66,36,24,60,231,255,
195,129,,,1,3,7,4,7,7,3,3,128,192,22
4,32,224,224,192
102 DATA192,7,71,77,221,253,39,35,12
```

```
7,224,226,178,187,191,228,196,254,,,
,24,24,,,,,
103 DATA36,24,24,36,,16,128,26,36,90
,90,36,154,2,60,66,129,153,153,129,6
6,60,,,36,16,,40,,
104 DATA,68,0,18,,,68,4,128,192,192,
225,251,255,255,255,,,64,224,248,252
,252,255,,,,,24
105 DATA60,126,255,,,,7,15,31,95,255
,,,,129,195,231,255,255
106 FORI=7168TO7327:READA:POKEI,A:NE
XT:A=8:POKE56,28:POKE52,28
107 FORI=7424TO7431:POKEI,0:NEXT:FOR
I=7328TO7335:POKEI,255:NEXT:POKE3686
9,255:GOTO490
108 PRINT"[CLR][BLU][22CRSRD]OPQRSOT
TTTTOPQRSOPQRS[HOM]"
110 B=PEEK(197):A=A-(B=37ANDA<18)+(B
=29ANDA>0):PRINT"[HOM][19CRSRD]"TAB(
A)A$
115 ONMGOTO150,150,200,200,300:M=INT
(RND(1)*6)
130 IFPEEK(653)=1ANDEN>50THENGOSUB20

135 PRINT"[HOM][RVSON][CYN]SCORE="SC
:PRINT"[HOM][RVSON][11CRSRR];FUEL="S
TR$(EN)"[SPC]"
137 IFEN<50THEN400
140 GOTO110
150 IFAU=0THENAU=1:P=7702+INT(RND(TI
)*22)
155 POKEP,32:E=P-INT((P-7680)/22)*22
-7682:P=P-22*(P<8120)-(E<A)+(E>A)
157 IFPEEK(P)<>32THEN400
158 IFPEEK(8100+E)=5ANDRND(1)<.5THEN
I=RND(1):P=P-(I<.5)+(I>=.5)
160 POKEP,2:POKEP+C,5:FORI=250TO125S
TEP-5:POKER1,I:NEXT:GOTO130
200 IFAU=0THENAU=1:P=7746+INT(RND(1)
*14)*22:I=RND(1):G=-(I<.5)+(I>.5):F%
=(G<0)+(P-7680)/22
210 POKEP,32:P=P+G:IFINT((P-7680)/22
)<>F%THENAU=0:M=0:GOTO130
220 POKEP,3:POKEP+C,4:FORSO=225TO250
:POKES1,SO:NEXT:POKES1,0:IFRND(TI)>=
.2THEN130
225 FORI=P+22TO8076+P-7680-22*INT((P
-7680)/22)STEP22:S=S+5:F=I
235 POKEI,1:POKEI+C,5:POKER1,128+S:N
EXT:S=0:IFPEEK(I+22)<>32THEN250
237 POKER1,0:S=0
240 FORI=P+22TOFSTEP22:POKEI,32:NEXT
:GOTO130
250 POKER1,0:FORU=9TO14:POKEI,U:POKE
R1,250-10*U:POKEI+C,CL:CL=14/CL:FORT
A=1TO10:NEXT:NEXT
255 POKER1,0:EN=INT(EN/2):GOTO240
300 IFAU=0THENAU=1:P=7704+INT(RND(1)
*18)
310 POKEP,32:P=P-22*(P<8142)-(P>8142
):POKEP,4:POKEP+C,1:IFP>8160THENPOKE
P,32:AU=0:M=0
320 FORI=200TO150STEP-2:POKES1,I:NEX
T:POKES1,0
330 IFPEEK(P+22)<>20THEN130
400 PRINT"[HOM][5CRSRD][RVSON][SPC]S
E[SPC]ACABO[SPC]SU[SPC]ENERGIA[SPC]!
":PRINT"[HOM][RVSON][11CRSRR][CYN];F
UEL=[SPC]0[3SPC]"
410 FORI=128TO250STEP2:POKER1,I:POKE
S1,I+5:FORT=1TO50:NEXT:NEXT:POKES1,0
:POKER1,0
420 PRINT"[RVSON][7CRSRD][4SPC]PULSE
[SPC]UNA[SPC]TECLA":POKE198,0
430 GETJ$:IFJ$=""THEN430
440 M=0:AU=0:A=8:EN=1000:NN=NN+1:IFN
N<3THEN108
450 PRINT"[CLR]":POKE36869,240:PRINT
"[YEL]LO[SPC]SIENTO,PERO[SPC]HAN[4SPC]
[CRSRD]GANADO[SPC]LOS[SPC][RED]INVAS
ORES."
455 PRINT"[CRSRD][GRN]LOGRASTE"SC"PU
NTOS."
460 PRINT"[CRSRD][YEL]QUIERE[SPC]JUG
AR[SPC]MAS(S/N)?":POKE198,0
470 GETJ$:IFJ$<>"S"ANDJ$<>"N"THEN470

480 IFJ$="N"THENPRINT"[GRN]PUES[SPC]
ADIOS.":END
490 POKE36869,255
500 PRINT"[CLR][YEL][RVSON]*****[SPC]
INVASORES[SPC]*****[CRSRD]DEBES[SPC]
DEFENDER[SPC]TU[SPC]PLA-[CRSRD]NETA[SPC]
DE[SPC]LA[SPC][PUR]INVASION[3SPC][CRSRD]
";
510 PRINT"GALACTICA[YEL][SPC]CON[SPC]
TU[SPC]BASE.[CRSRD]";
530 PRINT"[RVSON][YEL]TUS[SPC]ENEMIG
OS[SPC]SON:[5SPC][CRSRD][3SPC][RVSOFF]
[GRN]B[RVSON]:MUTANTE.":PRINT"[CRSRD]
[3SPC][RVSOFF][PUR]C[RVSON]:CAZA."
540 PRINT"[CRSRD][3SPC][RVSOFF][WHT]
D[RVSON]:TRANSPORTE.":PRINT"[YEL][RVSON]
[CRSRD]MANDOS:";
542 PRINT"[SPC][RVSON][RED]<:IZQDA."
:PRINT"[CRSRD][8SPC][RVSON][CYN]>:DC
HA.[SPC]":PRINT"[CRSRD][8SPC][RVSON]
[PUR]SHIFT:DISPARO."
545 PRINT"[RVSON][YEL]PULSA[SPC]UNA[SPC]
TECLA.[HOM]"
550 GETJ$:IFJ$=""THEN550
560 RESTORE:NN=0:GOTO108
```

Computer Bank



Es un juego derivado del gugol, en el que una entidad financiera realiza emisiones de series de seis billetes de distinto importe.

Cada vez que un jugador va a realizar una jugada deberá aportar al fondo o poceta de apuestas la cantidad acordada inicialmente.

Al jugador a quien corresponde jugar se le muestran uno a uno los billetes de la serie. El deberá procurar quedarse con el de

valor más alto, no pudiendo ejercer ninguna opción sobre los que rechazó con anterioridad al billete que se le muestra. Si su elección recae sobre uno de los cinco primeros obliga a jugar sobre el último de la serie, caso de que este sea el mayor de su serie deberá repartir la poceta con el resto de los jugadores. Si un jugador rechaza el quinto billete, está obligado a aceptar el sexto. Agotada la serie, si algún jugador ha elegido el mayor, ha ganado esa serie y deberá retirar el importe de las apuestas acumuladas hasta ese momento en la poceta.

Los valores de los billetes que componen una serie son generados aleatoriamente por el ordenador.

El diagrama de flujo y el cuadro de variables ayudan a comprender el funcionamiento del programa. A continuación se realiza una breve descripción línea a línea.

La línea número 10 establece los valores de las variables HT$ y VT$ destinadas en el programa a tareas de tabulación horizontal (H) y vertical (V).

La siguiente línea realiza las siguientes funciones:

a) Borra la pantalla y establece el color negro como habitual para mensajes mostrados a través de pantalla.

b) Inicializa el contenido de la variable CL$ (colores de billetes emitidos).

c) Asigna valor inicial a la variable VD (velocidad de interpretación de melodías). El valor inicial sólo se mantiene durante la presentación.

d) Reserva espacio de memoria para la matriz MU% (6,10,3) destinada a contener los valores correspondientes a 6 melodías musicales de 10 notas como máximo. Cada nota se caracteriza por sus valores de ALTA, BAJA y DURACION. Los valores están en sentencias DATA correlativas a partir de la línea 40420.

La línea 50 llama a la subrutina 22000 que realiza las siguientes funciones:

1) Llama a la subrutina 23000 que inicializa los valores del sintetizador.

2) Llama a la subrutina 24000 que:

a) Borra pantalla.

b) Llama a la subrutina 21000 que genera una serie aleatoria de seis billetes.

c) Muestra los seis billetes en pantalla. Esta función la realiza la subrutina 20000 dentro de un lazo controlado por la variable NB (número de billete).

d) Lee e interpreta las 32 notas colocadas en las sentencias data de las líneas 40000 - 40420 inclusive.

e) Multiplica por 5 la velocidad de las melodías restantes que se ejecutarán a lo largo del programa dividiendo, para ello la variable VD (inversa a la velocidad de interpretación de melodías).

3) Borra el contenido de la pantalla.

Las líneas 100 a 300 muestran las instrucciones del juego.

La línea 1000 sirve para introducir el valor en pesetas de la apuesta básica. El valor asumido por defecto es de cinco pesetas.

La línea 1010 permite indicar el número de jugadores (variable NJ). El valor por defecto es de dos. El juego carece de sentido para un solo jugador ya que al final de la partida su ganancia neta, repartiendo la poceta sería de cero.

El programa utiliza la línea número 1020 para solicitar los nombres de los jugadores que participan en el juego.

La línea 1030 permite la construcción de un literal TJ$ que contiene los valores ascii de los turnos de juego.

La línea 1040 llama a la subrutina 21000 que como hemos visto genera aleatoriamente una serie de seis billetes e incrementa en una unidad al contador JU que contiene a lo largo de la partida el número de jugada, esto es, el número de la emisión realizada.

La siguiente línea realiza los cometidos que se indican:

a) Inicia el contador de billetes NB.

b) Calcula el valor de la variable TJ (turno de juego), a partir del cual deberemos posicionarnos en el string TJ$ para examinar el orden de actuación de los distintos jugadores en la jugada número JU.

c) Borra pantalla y coloca en la segunda línea en inverso el rótulo del juego "computer bank".

d) Inicia el contador JG correspondiente al jugador ganador.

La línea 1080 inicia el lazo de todos los jugadores que pueden tener opción a apostar sobre la serie emitida número JU.

La línea 1090 incrementa en una unidad el contador de billetes (NB: número de billete sobre el que se tendrá opción en ese momento) y si el billete es el último se bifurcará a la línea 1410 ya que de acuerdo con las reglas de juego convenidas el jugador deberá aceptarlo obligatoriamente.

La siguiente línea establece el número de jugador a quien corresponde jugar, mostrando por pantalla su nombre.

La línea número 1110 realiza las siguientes operaciones:

a) Coloca el puntero I2 en 1. Número de la melodía que debe interpretarse.

b) Coloca el puntero I3 en 4. Número de notas de la melodía número I2 que deben interpretarse.

c) Llama a la subrutina 25000 que interpretará la melodía 12 de I3 notas.

La línea 1120 inicia un lazo que se realiza dos veces y que incluye:

a) Mostrar mensaje "coloque 5 pesetas" o cualquier otra cantidad elegida como apuesta básica.

b) Interpretar la 5 melodía de 5 notas.

c) Borrar mensaje.

d) Retardo controlado por la variable J (ver línea 1140).

La línea 1140, además de las funciones correspondientes al bucle iniciado en la línea 1120, actualiza el valor de la poceta y visualiza en pantalla un mensaje que indica su contenido actual.

La línea 1145 descuenta de la cuenta del jugador que juega PJ el valor de la apuesta. De esta forma se acumula la variable TT (PJ) las pérdidas que le ocasiona al jugador PJ las cantidades que coloca en el fondo o poceta.

La línea 1150 visualiza el billete NB, llamando para ello a la subrutina número 20000, e interpreta la melodía número 4 de 6 notas, cediendo el control a la subrutina 25000.

La línea 1160 visualiza mensaje indicando al jugador que éste es el momento en el que puede decidir si acepta o no el billete que se le muestra. La respuesta del jugador es la contenida en la variable RE$.

Si la respuesta es negativa (línea 1170) se bifurcará a la 1400 donde se comprueba si quedan o no billetes ya que de quedar sólo uno deberá continuar en secuencia en la 1410 ya que el jugador está obligado a aceptar el último de la emisión; caso contrario se incrementa el contador de billetes visualizando el siguiente, para lo cual bifurca a la línea 1150 donde se repite el proceso ya descrito.

Si la respuesta no ha sido negativa y tampoco lo es afirmativa (línea 1180) la entrada ha sido incorrecta por lo que se cede el control a la línea 1160 insistiendo de esta forma en la pregunta hasta que se produzca una respuesta válida (S o N).

La secuencia a partir de la línea 1190 se realiza cuando el billete es aceptado. La primera de estas líneas comprueba si el valor del billete mostrado es el mayor de la serie en cuyo caso JG (jugador ganador) es el que acaba de aceptar ese billete, jugador PJ.

La siguiente línea cierra el lazo de los jugadores que tienen opción a jugar sobre esa emisión de billetes.

Si llegan a jugar todos los jugadores que tienen opción sobre una determinada serie de billetes el programa se encontrará en la línea 1210, donde comprueba si el ganador está desierto (JG=0) en cuyo caso interpreta la melodía 3 de 10 notas y bifurca a la línea 1240 en la que, como veremos, se plantea la continuidad de la partida.

Si hay ganador debe acumular en su cuenta las ganancias obtenidas (1220) e interpreta la melodía 2 de cuatro años. El contenido final de la cuenta del jugador n será TT(n) = ganancias acumuladas — apuestas realizadas = ganancias netas (o pérdidas si el valor final es negativo).

La línea 1220 visualiza mensaje indicador para que el jugador ganador retire la poceta y, en ese momento hace el valor de la poceta igual a cero.

La siguiente instrucción, contenida en la línea 1240 plantea la continuidad de la partida. Si la respuesta es afirmativa se bifurca a la línea 1040 en la que se genera una nueva serie de billetes (ver línea 1250)

Si la respuesta no ha sido afirmativa y tampoco es negativa, ha

sido una respuesta no valida insistiendo de nuevo acerca de la continuidad de la partida (ver línea 1260).

Si la respuesta ha sido negativa se visualizan los resultados, mediante un rótulo (1270), indicación del valor actual de la poceta que debe repartirse entre todos los jugadores que han participado en el juego (línea 1280), y las ganancias de cada uno de ellos (pérdidas o ganancias netas) emitiendo un mensaje genérico línea 1290) y uno específico para cada jugador (línea 1310), finalizando el programa.

Si un jugador se ve obligado a aceptar el último billete de la serie, se visualiza emitiendo la melodía característica y si no es el mayor, se bifurca a la línea 1210 donde se comprobará la existencia o no de ganador.

Si es ganador el sexto billete se calcula la cuota de reparto de la poceta (K). Si esa cuota es el total es que sólo gana un jugador (1420).

La línea 1430 visualiza el monto al que asciende el reparto de la poceta.

La línea siguiente emite la melodía correspondiente al reparto de la poceta y realiza las sumas en las cuentas de los jugadores afectados.

La línea 1450 calcula el estado en que queda el fondo tras el reparto (resto) y bifurca a la línea 1210, en la que se plantea la cuestión de continuidad de la partida.

Nota: En un listado posterior se han agrupado sentencias data.

LISTA DE VARIABLES:

HT$ VT$	Tabulación
CL$	Colores
RE$	Comunicación con el operador
PT	Unidad monetaria de apuesta en pesetas
NJ	Número de jugadores
N$(i)	Nombre jugador i. $1 \leqslant i \leqslant NJ$
T$	Turnos de juego
TJ	Turno inicial en cada serie de billetes emitida
JA	Nº de orden del jugador que apuesta
JP	Indice del jugador que apuesta
JG	Jugador ganador
MA	Número de orden del mayor billete de cada serie
K	Cantidad a repartir de la poceta
BI (i)	Importe del billete de orden i
BI$	String cantidad del billete que se visualiza
TP	Cantidad más alta posible dentro de una serie de billetes
JU	Número de jugadas enteras=nº series emitidas
TT (i)	Pérdidas o ganancias netas del jugador i
PO	Importe del fondo en un momento dado (poceta)
I2, I3	Punteros correspondientes a número de melodía y número de notas de la misma.

Las melodías han sido compuestas para este juego por mi cuñado José Bermejo.

```
1 REM COMPUTER BANK
10 HT$="[HOM][25CRSRD]":VT$="[39CRSRR]
"
20 PRINT"[CLR][BLK]":CL$="[BLK][CYN]
[PUR][GRN][YEL][COMM6]":VD=60:DIM MU
%(6,10,3)
50 GOSUB 22000
100 PRINT"[2SPC]EL[SPC]JUEGO[SPC]CON
SISTE[SPC]EN[SPC]ELEGIR[SPC]EL[SPC]B
ILLETEMAS[2SPC]ALTO[SPC]DE[SPC]CADA[SPC]
SERIE[SPC]DE[SPC]6";
110 PRINT"[2SPC]QUE[2SPC]EMITE[RVSON]
COMPUTER[SPC]BANK[RVSOFF]."
120 PRINT"[CRSRD][2SPC]PARA[SPC]ELLO
[SPC]SE[SPC]LE[SPC]MUESTRAN[SPC]UNO[2SPC]
A[2SPC]UNO[2SPC]PUDIENDO[SPC]ELEGIR[SPC]
EL[SPC]QUE,[SPC]EN";
130 PRINT"[SPC]CADA[SPC]MOMENTO,SE[SPC]
LE[SPC]MUESTRA."
140 PRINT"[CRSRD][2SPC]SI[SPC]UD[SPC]
RECHAZA[SPC]LOS[SPC]CINCO[SPC]PRIMER
OS[2SPC]SE[2SPC]VERA[SPC]OBLIGADO[SPC]
A[SPC]ACEPTAR[SPC]";
150 PRINT"EL[SPC]ULTIMO.[2SPC]EN[2SPC]
ESTE[SPC]CASO,[CRSRR]DE[SPC]GANAR,[SPC]
REPARTIRA[SPC]LA[SPC]POCETACON[SPC]L
OS[SPC]";
160 PRINT"OTROS[SPC]JUGADORES."
170 PRINT"[CRSRD][2SPC]SI[SPC]UD[SPC]
ELIGE[SPC]ALGUNO[SPC]DE[SPC]LO[SPC]C
INCO[2SPC]PRIME-ROS,EL[SPC]SIGUIENTE
[SPC]JUGADOR";
180 PRINT"[SPC]TENDRA[SPC]OPCION[SPC]
AELEGIR[SPC]EL[SPC]MAYOR[SPC]DE[SPC]
ESA[SPC]EMISION[2SPC]GANANDO[SPC]LA[SPC]
POCETA."
190 PRINT"[CRSRD][2SPC]INDEPENDIENTE
MENTE[SPC]DE[SPC]LO[SPC]SENALADO[SPC]
EN[2SPC]LA[SPC]REGLA[SPC]ANTERIOR,[SPC]
SOLO[SPC]";
200 PRINT"CORRERA[SPC]EL[SPC]TURNODE
[SPC]JUEGO[SPC]CADA[SPC]VEZ[SPC]QUE[SPC]
SE[SPC]AGOTE[SPC]UNA[SPC]NUEVAEMISIO
N."
210 PRINT"[RVSON][6SPC]PULSE[SPC]UNA
[SPC]TECLA[SPC]PARA[SPC]CONTINUAR[4SPC]
[RVSOFF]";
220 GET RE$:IF RE$=""THEN220
230 PRINT"[CLR][CRSRD][2SPC]ANTES[SPC]
DE[SPC]INICIAR[SPC]EL[SPC]JUEGO[SPC]
PROPIAMENTE[SPC]DICHO[SPC]DEBERA[SPC]
INTRODUCIR[SPC]";
240 PRINT"LA[SPC]UNIDAD[SPC]MONE-[SPC]
TARIA[SPC]QUE[SPC]SE[SPC]APUESTA[SPC]
CADA[SPC]VEZ[2SPC]QUE[2SPC]SE[2SPC]T
IENE[SPC]OPCION";
250 PRINT"[SPC]DE[SPC]ELEGIR[SPC]UN[SPC]
BILLETE.[SPC]"
260 PRINT"[CRSRD][2SPC]TAMBIEN[SPC]D
EBE[SPC]INDICAR[SPC]LOS[SPC]NOMBRES[SPC]
DE[3SPC]LOS[SPC]JUGADORES[SPC]A[SPC]
FIN[SPC]DE[SPC]QUE";
270 PRINT"[SPC]LA[2SPC]MAQUINA[2SPC]
LLEVE[SPC]EL[SPC]SISTEMA[SPC]DE[SPC]
TURNOS[SPC]Y[SPC]LA[SPC]CONTABI-LIDA
D[SPC]DE[SPC]LA";
280 PRINT"[SPC]PARTIDA[SPC]([SPC]POC
ETA[SPC]Y[SPC]GANANCIASNETAS[SPC]).
290 PRINT"[5CRSRD][RVSON][6SPC]PULSE
[SPC]UNA[SPC]TECLA[SPC]PARA[SPC]CONT
INUAR[4SPC][RVSOFF]";
300 GET RE$:IF RE$=""THEN300
1000 INPUT"[CLR][2CRSRD]UNIDAD[SPC]D
E[SPC]APUESTA[SPC]EN[SPC]PTAS:[2SPC]
5[3CRSRL]";PT
1010 INPUT"NUMERO[SPC]DE[SPC]JUGADOR
ES:[2SPC]2[3CRSRL]";NJ
1020 FOR I=1 TO NJ:PRINT"NOMBRE[SPC]
JUGADOR[SPC]"I:INPUT N$(I):NEXTI
1030 TJ$=CHR$(NJ):FOR I=1 TO NJ :TJ$
=TJ$+CHR$(I):NEXT I:TJ$=TJ$+TJ$
1040 GOSUB 21000:JU=JU+1
1070 NB=0:TJ=JU-NJ*INT(JU/NJ):PRINT"
[CLR][CRSRD]"TAB(13)"[RVSON][SPC]COM
PUTER[SPC]BANK[SPC]":JG=0
1080 FOR JA=1+TJ TO NJ+TJ
1090 NB=NB+1:IFNB>5THEN 1410
1100 PJ=ASC(MID$(TJ$,JA,1)):PRINT LE
FT$(HT$,23)"JUEGA[SPC]"N$(PJ)"[18SPC]
"
1110 I2=1:I3=4:GOSUB 25000
1120 FORI=1TO2 :PRINT LEFT$(HT$,23)"
COLOQUE[2SPC]"STR$(PT)"[SPC]PTAS.[12SPC]
"
1130 I2=5:I3=5:GOSUB 25000:PRINT LEF
T$(HT$,23)"[30SPC]"
1140 FORJ=1TO300:NEXTJ,I:PO=PO+PT:PR
INT LEFT$(HT$,22)"[RVSON][SPC]POCETA
[SPC]"STR$(PO)"[SPC]PTAS.[RVSOFF]"
1145 TT(PJ)=TT(PJ)-PT
1150 GOSUB 20000:I2=4:I3=6:GOSUB 250
00
```

```
1160 PRINT LEFT$(HT$,23)"LO[SPC]TOMA
[SPC](S/N)[10SPC]";:INPUT "S[3CRSRL]
";RE$
1170 IF RE$="N" THEN 1400
1180 IF RE$<>"S"THEN 1160
1190 IF NB=MA THEN JG=PJ
1200 NEXT JA
1210 IF JG=0 THEN I2=3:I3=10:GOSUB 2
5000:GOTO1240
1220 TT(JG)=TT(JG)+PO:I2=2:I3=4:GOSU
B 25000
1230 PRINT LEFT$(HT$,23)"GANA[SPC]"N
$(JG)"[SPC]Y[SPC]RETIRA[SPC]LA[SPC]P
OCETA[10SPC]":PO=0
1240 INPUT"DESEAN[SPC]CONTINUAR[3SPC]
S[3CRSRL]";RE$
1250 IF RE$="S"THEN 1040
1260 IF RE$<>"N"THEN 1240
1270 PRINT"[CLR][CRSRD]"TAB(14)"[RVSON]
RESULTADOS"
1280 PRINT"[3CRSRD]SE[SPC]REPARTE[SPC]
UNA[SPC]POCETA[SPC]DE[SPC]"PO"PTAS"
1290 PRINT"[CRSRD]SALVO[3SPC]ELLO[2SPC]
SE[2SPC]HAN[SPC]ACUMULADO[3SPC]LAS[38PC]
SIGUIENTES[SPC]GANANCIAS"
1310 FOR I=1 TO NJ:PRINTN$(I),TT(I):
NEXTI: END
1400 IF NB<5 THEN NB=NB+1:GOTO 1150
1410 NB=6:GOSUB 20000:I2=4:I3=6:GOSU
B 25000:IF MA<>6 THEN 1210
1420 K=INT(PO/(TJ+NJ-JA+1)):IF K=PO
THENJG=PJ:GOTO1220
1430 PRINT"[CLR]SE[SPC]REPARTEN[SPC]
LA[SPC]POCETA[SPC]TOCANDO[SPC]A[SPC]
"K
1440 I2=6:I3=7:GOSUB 25000:FOR I=1 T
O NJ:PRINT N$(I):TT(I)=TT(I)+K:NEXT
I
1450 PO=PO-K*(TJ+NJ-JA+1):GOTO 1240
20000 REM MOSTRAR BILLETE NB DE BI(N
B)PTESETAS
20010 PH=INT((NB-1)/3):PV=NB-1-3*INT
((NB-1)/3)
20020 PRINT LEFT$(HT$,5+9*PH)TAB(1+1
3*PV);LEFT$(CL$,RND(0)*6+1);
20030 FOR I=1 TO 2:PRINT"[RVSON][12SPC]
[RVSOFF][12CRSRL][CRSRD]";:NEXT I
20040 PRINT"[RVSON][2SPC]COMPUTER[2SPC]
[RVSOFF][12CRSRL][CRSRD]";
20050 PRINT"[RVSON][4SPC]BANK[4SPC][RVSOFF]
[12CRSRL][CRSRD]";
20060 BI$=STR$(BI(NB)):BI$=RIGHT$(BI
$,LEN(BI$)-1)
20070 BI$=RIGHT$("[4SPC]",(12-LEN(BI
$))/2)+BI$
20080 FORI=LEN(BI$)TO 11:BI$=BI$+"[SPC]
":NEXT I
20090 PRINT"[RVSON]"BI$"[RVSOFF][12CRSRL]
[CRSRD]";
20100 FOR I=1 TO 2:PRINT"[RVSON][12SPC]
[RVSOFF][12CRSRL][CRSRD]";:NEXT I:PR
INT"[BLK]":RETURN
21000 TP=INT(9999*RND(0))
21010 MA=0:FOR I=1 TO 6:BI(I)=INT((1
+TP*RND(0))):IF BI(I)>BI(MA)THENMA=I
21020 NEXT I:RETURN
22000 GOSUB 23000:GOSUB 24000
22100 PRINT"[CLR]"
22999 RETURN
23000 S=54272:FORL=0TO 24:POKE S+L,0
:NEXT L:POKE S+3,8
23010 POKE S+5,41:POKE S+6,89:POKE S
+14,117:POKE S+18,16:POKE S+24,143
23020 RETURN
24000 PRINT"[CLR]":GOSUB 21000:FOR N
B=1 TO 6:GOSUB 20000:NEXT NB:FORI2=1
 TO 32
24010 READ HF,BF,DF
24020 POKE S+4,65
24030 POKE S+0,BF:POKES+1,HF
24040 FOR T5=1 TO VD*DF:NEXT T5:NEXT
 I2
24050 FOR I2=1 TO 6:FOR I3=1 TO10:FO
R I4=1 TO 3:READ MU%(I2,I3,I4):NEXT
I4,I3,I2
24060 VD=VD/5:RETURN
25000 FORI5=1 TO I3
25010 HF=MU%(I2,I5,1):BF=MU%(I2,I5,2
):DF=MU%(I2,I5,3)
25020 POKE S+4,65
25030 POKE S+0,BF:POKES+1,HF
25040 FOR T5=1 TO VD*DF:NEXT T5:NEXT
 I5
25050 POKE S+0,0:POKES+1,0
25060 RETURN
40000 DATA 12,216,3,14,107,1,16,47,4
,12,216,4,11,114,3,12,216,1,14,107,4
,11,114
40070 DATA 4,10,205,3,11,114,1,12,21
6,4,10,205,4,9,159,3,8,147,1,12,216,
4,0,0,4
40160 DATA 12,216,3,14,107,1,16,47,4
,12,216,4,11,114,3,12,216,1,14,107,4
,11,114
40170 DATA 4,10,205,3,11,114,1,12,21
6,3,10,205,1,9,159,4,12,216,4,8,147,
8,0,0,0
40420 DATA 8 ,147,4,10,205,4,12,216,
4,17,37,12,0,0,0,0,0,0,0,0,0,0,0,0
,0,0
40490 DATA 0,0,0,12,216,9,11,114,3,1
0,205,6,8,147,6,0,0,0,0,0,0,0,0,0,
0,0
40580 DATA 0,0,0,0,0,0,12,216,3,11,1
14,3,12,216,3,11,114,3,10,205,4,9,15
9,4
40660 DATA 10,205,4,8,147,6,8,23,6,8
,147,12,12,216,3,15,70,3,12,216,6,12
,216,3
40740 DATA 15,70 ,3,12,216,6,0,0,0,0
,0,0,0,0,0,0,0,0,0,8,147,12,9,159,12,1
0,205,12
40830 DATA 11,114,12,12,216,12,0,0,0
,0,0,0,0,0,0,0,0,0,0,8,147,6,10,
205,6
40920 DATA 9 ,159,6,11,114,6,10,205,
12,9,159,12,8,147,12,0,0,0,0,0,0,0,0
,0
```



En una línea

He aquí uno de los mejores programas de una línea que funciona tanto para el VIC-20 como para el C-64 y que dibuja un laberinto continuo.

```
8 Print chr$(205.5+rnd(8));:goto8
```

Para conseguir también colores aleatorios en el C-64, hay que poner CHR$(149+rnd(8)*11) antes del punto y coma.

D.A.K. RUN EEUU.

Números aleatorios

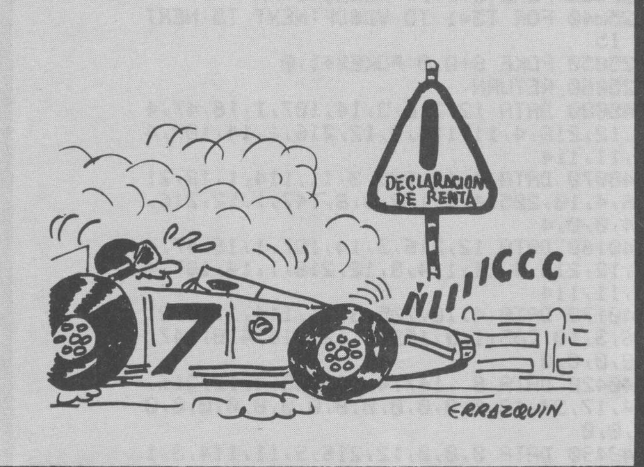
Algunos programas usan repetidas veces sentencias del tipo: X=INT(N*RND(0))+M. Se puede ahorrar bastante tiempo usando: X%=N*RND(.)+M en vez de lo anterior, el usar una variable entera hace el INT y el punto sustituye al 0, ganando bastante tiempo.

S.A.B. RUN EEUU.

Fórmula 1

José Luis Armenteros León (13 años)
C/Luis de Camoens, 4 - piso 5º
Telf.: 26 10 03
La Coruña

VIC-20



Consiste en manejar a un coche blanco por una carretera (con las teclas Z y .) esquivando a los coches amarillos. Pulsando una tecla al final, se ve la puntuación y el juego vuelve a empezar.

```
10 POKE650,128:P=0
20 POKE52,28:POKE56,28
30 FORX=7168TO7679:POKEX,PEEK(X+2560
0):NEXTX
40 POKE36869,255
50 FORX=7168TO7175:READD:POKEX,D:NEX
TX
60 FORX=7176TO7183:READD:POKEX,D:NEX
TX
70 FORX=7184TO7191:READD:POKEX,D:NEX
TX
80 POKE36879,15:PRINT"[CLR]"
90 B=7:A=7690
95 PRINT"NIVEL[SPC]DEL[SPC]JUEGO"
96 INPUT"15=DIFICIL[SPC]150=FACIL";C
:POKE36879,10
97 IFC<15ORC>150THEN96
98 FORL=1TO50:PRINTTAB(B)"A[7SPC]A":
NEXT
100 X=INT(RND(1)*C)+1
110 IFX>14THENPRINTTAB(B)"A[7SPC]A":
GOTO290
120 IFX=1THENPRINTTAB(B)"A[7SPC]A":G
OTO290
130 IFX=2THENPRINTTAB(B)"AA[6SPC]AA"
:B=B+1:GOTO290
140 IFX=3THENB=B-1:PRINTTAB(B)"AA[6SPC]
AA":GOTO290
150 IFB<1THENB=1
160 IFB>11THENB=11
170 POKE36878,0:POKE36878,15
180 IFX=4THENPRINTTAB(B)"A[YEL]B[BLU]
[6SPC]A":GOTO290
190 IFX=5THENPRINTTAB(B)"A[SPC][YEL]
B[BLU][5SPC]A":GOTO290
200 IFX=6THENPRINTTAB(B)"A[2SPC][YEL]
B[BLU][4SPC]A":GOTO290
210 IFX=7THENPRINTTAB(B)"A[3SPC][YEL]
B[BLU][3SPC]A":GOTO290
220 IFX=8THENPRINTTAB(B)"A[4SPC][YEL]
B[BLU][2SPC]A":GOTO290
230 IFX=9THENPRINTTAB(B)"A[5SPC][YEL]
B[BLU][SPC]A":GOTO290
240 IFX=10THENPRINTTAB(B)"A[6SPC][YEL]
B[BLU]A":GOTO290
250 IFX=11THENPRINTTAB(B)"A[YEL]B[5SPC]
B[BLU]A":GOTO290
260 IFX=12THENPRINTTAB(B)"A[SPC][YEL]
B[3SPC]B[BLU][SPC]A":GOTO290
270 IFX=13THENPRINTTAB(B)"A[2SPC][YEL]
B[SPC]B[BLU][2SPC]A":GOTO290
280 IFX=14THENPRINTTAB(B)"A[2SPC][YEL]
BBB[BLU][2SPC]A"
290 E=E+1:IFE=10THENE=0:C=C-1:IFC<14
THENC=14
300 GETC$:IFC$="Z"THENA=A-1
310 IFC$="."THENA=A+1
320 POKEA,0:POKEA+30720,1:P=P+1
330 IFPEEK(A+22)<>32THEN400
340 POKE36875,125:POKE36875,225
350 GOTO100
360 DATA189,231,189,36,165,255,153,6
0
370 DATA66,66,66,66,66,66,66,66
380 DATA60,153,255,165,36,189,231,18
9
400 POKE36877,220:FORL=15TO0STEP-1:P
OKE36878,L
410 FORM=1TO300:NEXTM:NEXTL
420 POKE36877,0:POKE36878,0
430 GETD$:IFD$=""THEN430
435 PRINT"PUNTOS=";P
440 RESTORE:GOTO10
```



Subrutina para centrar textos

Supongamos que tenemos una cadena llamada A$, y queremos centrarla en la pantalla. Sólo hay que hacer una llamada a la siguiente subrutina:

```
1000 PRINT TAB (40-LEN(A$)/2);A$:RETURN
```

Si en vez de un 64 tienes un VIC-20, cambia el valor 40 por el de 22, que es el ancho de la pantalla del VIC.

M.B. RUN EEUU.

Bucles de espera

Una manera útil de programar unos bucles de espera es la siguiente:

```
100   fort=1to100:next:return
1000 fort=1to1000:next:return
1500 fort=1to1500:next:return
etc.
```

Cuando deseemos que el programa realice una espera, escribimos: GOSUB XXXX, donde XXXX es el tiempo de espera. Por supuesto la línea XXXX debe existir.

W.W.B. RUN EEUU

Letra a letra

Para imprimir textos letra a letra y lentamente, escribe:

```
10 a$="este es el mensaje":gosub1000
20 end

1000 fori=1to len(a$):Print mid$(a$,i,1);:for t=1 to
50:nextt,i:return
```

Para conseguir diferentes velocidades, basta con cambiar el valor de t en el for-next.

C.B. RUN EEUU.

3-R. Functions

Pablo Cambra Brown
Martínez Valls, 14. P3
Onteniente (Valencia)

C-64

Este programa lo que hace es representar funciones en la alta resolución del Commodore 64.

Tal vez ésta sea la más complicada de explicar, pero trataré de explicarlo lo mejor que pueda, aunque la mejor manera de entenderlo es con la práctica.

Antes de correr el programa es fundamental conocer los puntos de discontinuidad* de la función, o sea, donde (para que valor de X) la función se hace infinito. Si no sabe calcular estos valores será mejor olvidarse del programa por completo. Cito a Leonardo da Vinci: "Que no lea mis principios quien no sea matemático".

El programa en un principio nos pide la función a representar, esta se introduce vía un INPUT.

Posteriormente se apreta RETURN 3 veces y el programa pide la escala y los puntos.

A estas alturas os habréis dado cuenta que el programa está íntegramente escrito en inglés, esto es debido a que las funciones como el seno, coseno, logaritmo, etc., sólo son utilizables en el ordenador con sus abreviaturas inglesas, y ya puestos, ya que hay que escribir las funciones en inglés, que todo sea en inglés.

Como decía, la máquina nos pide la escala y los puntos. La escala sirve para determinar el tamaño de la función en la pantalla. Por ejemplo: si en la escala ponemos el número 20, los ejes (que ocupan toda la pantalla, de punta a punta y de arriba abajo) correrán desde el -8 hasta el 8 en el eje de las Xs y de -5 a 5 en el de las Yes, si la escala es 10, la función la mitad de tamaño del caso anterior y por tanto caben el doble de valores (es decir, de -16 a 16 en X, de -10 a 10 en Y), etc.

Los puntos es, digamos, lo que calibra la precisión de la representación: cuanto menor sea el parámetro (número) introducido más puntos de la función calculará. Esto se hace evidente cuando se mira la función que hace dicho número en el programa (es al valor de incremento —STEP— de X).

Una vez introducidos estos valores el ordenador quiere saber si deseamos comparar la función con otra que anteriormente hemos representado. Interesante para estudiar la fórmula de Taylor. Conteste YES o No.

En caso de contestar NO a la pregunta anterior, el ordenador le preguntará si desea que dibuje ejes para la representación, en caso de que la función se confunda con estas durante un trecho, recomiendo que conteste NO. Si este es el caso, el ordenador representará los puntos de los ejes, pero no éstos. En caso contrario, el ordenador colocará los ejes y sus puntos.

Lo siguiente que pide el ordenador es el color de la pantalla. Si tiene una "tele" b/n recomiendo aquí el color 4, que es más o menos neutro. Otro color recomendable es el 16 (blanco sobre negro), pero se ocasionan distorsiones en caso de no tener monitor.

Ahora llegamos a lo peliagudo de la cuestión y aquí entran en juego los puntos de discontinuidad previamente calculados. Imaginemos que hemos elegido como escala 20 y hay un punto de discontinuidad en X=-3 y en ningún otro sitio. Escribiremos como respuesta a la pregunta de: HOW MANY INTERVALS? 2 (porque debido a la discontinuidad en un solo punto, la función se "divide" en dos). Después viene el "de dónde a dónde)" quiere que vaya dicha función. En el caso anterior se debería escribir: -8, -3.01 es decir parando justo antes de producirse la discontinuidad. Y al aparecer de nuevo el mensaje: "A,B?" (pidiendo los parámetros para el segundo intervalo) se escribiría: -2.99,8 es decir, justo después del punto donde se produce la discontinuidad hasta el final. Como se puede apreciar, el uso correcto de los intervalos y de sus parámetros, evita desagradables fallos de cálculo del ordenador (como un "?DIVISION BY ZERO ERROR IN XXX").

Después de introducir estos últimos números puede recostarse en su sillón mientras observa cómo la máquina realiza la representación gráfica que usted ha elegido.

Una vez finalizada la función, el ordenador le advertirá con un letrero luminoso.

Apriete RUN/STOP y RESTORE para volver a empezar.

* Discontinuidades de salto infinito.

```
1 CLR
2 INPUT"[CLR]FUNCTION";FU$:PRINT"[2CRSRD]
60[SPC]Y=INT(SC*("FU$")+.5)"
3 PRINT"4[SPC]F1$="CHR$(34)"Y="FU$CH
R$(34):PRINT"RUN4[6CRSRU]":END
4 F1$="Y=SIN(X)"
5 E=2.7182818
6 INPUT"[CLR]SCALE,DOTS";SC,DT:SC=AB
S(SC):DT=ABS(DT)
7 INPUT"COMPARE(YES/NO)[5SPC][5CRSRL]
";CP$
8 IFCP$<>"YES"ANDCP$<>"NO"THENPRINT"
[2CRSRU]":GOTO7
10 IFCP$="YES"GOTO14
12 INPUT"AXIS[SPC](YES/NO)[5SPC][5CRSRL]
";AX$
13 IFAX$<>"YES"ANDAX$<>"NO"THENPRINT
"[2CRSRU]":GOTO12
14 INPUT"PAPER[SPC]COLOUR";CL
15 INPUT"HOW[SPC]MANY[SPC]INTERVALS"
;DI:DIMA(DI),B(DI)
16 FORU=1TODI:INPUT"A[SPC]&[SPC]B";A
(U),B(U):NEXT:PRINT"[CLR]"
20 BS=8192:POKE53272,PEEK(53272)OR8
22 POKE53265,PEEK(53265)OR32
25 IFCP$="YES"THENFU$="[16SPC]":O=81
92:GOSUB3000:GOTO35
30 FORI=8192TO16911:POKEI,0:NEXT
35 IFCL=0GOTO42
40 FORI=1024TO2023:POKEI,CL:NEXT
42 IFCP$="NO"THENGOSUB1000
44 O=8192:FU$=F1$:GOSUB3000
45 O=15872:FU$="[8SPC]":GOSUB3000
46 FORT=1TODI
50 FORX=A(T)TOB(T)STEPDT/10
60 Y=INT(SC*(SIN(X))+.5)
64 IFY<-100ORY>100THEN150
66 IFX<-160/SCORX>160/SCTHEN150
68 AY=-Y
70 IX=(X*SC)+160
80 IY=AY+100
90 CH=INT(IX/8)
100 RO=INT(IY/8)
110 LN=IYAND7
120 BY=BS+RO*320+8*CH+LN
```

```
130 BI=7-(IXAND7)
140 POKEBY,PEEK(BY)OR(2↑BI)
150 NEXTX,T
155 O=15872:FU$="FINISHED":GOSUB3000

160 FORI=1TO16STEP15:FORU=1984TO1991
:POKEU,I:NEXT:FORJ=0TO30:NEXTJ,I:GOT
O160
1000 IFAX$="NO"THENS1=103:S2=157:AMT
=6:GOTO1255
1010 S1=99:S2=161:AMT=2
1050 FORX=-159TO159
1060 CH=INT(X/8)
1090 BY=BS+4000+8*CH+4
1100 BI=7-(XAND7)
1110 POKEBY,PEEK(BY)OR(2↑BI)
1120 NEXT
1130 FORY=0TO199
1200 RO=INT(Y/8)
1210 LN=YAND7
1220 BY=BS+RO*320+160+LN
1240 POKEBY,PEEK(BY)OR128
1250 NEXT
1255 IFSC<=1THENRETURN
1257 FORU=0TOAMT
1260 FORX=160TO319STEPINT(SC)
1270 Y=S1-U
1275 IX=X:FORI=0TO1
1280 CH=INT(IX/8)
1290 RO=INT(Y/8)
1300 LN=YAND7
1310 BY=BS+RO*320+8*CH+LN
1320 BI=7-(IXAND7)
1330 POKEBY,PEEK(BY)OR(2↑BI)
1340 IX=320-X:NEXTI,X
1360 FORY=100TO199STEPINT(SC)
1370 X=S2+U
1375 IY=Y:FORI=0TO1
1380 CH=INT(X/8)
1390 RO=INT(IY/8)
1400 LN=IYAND7
1410 BY=BS+RO*320+8*CH+LN
1420 BI=7-(XAND7)
1430 POKEBY,PEEK(BY)OR(2↑BI)
1440 IY=200-Y:NEXTI,Y,U:RETURN
3000 REM-PUESTA EN PANTALLA DE LA FU
NCION
3010 CR=1
3020 IFCR=LEN(FU$)+1THENRETURN
3025 J$=MID$(FU$,CR,1):CR=CR+1
3030 J=ASC(J$):IFJ>63THENJ=J-64
3040 V=53248+J*8
3050 POKE56334,0:POKE1,51
3060 FORI=1TO8
3070 POKEO,PEEK(V)
3080 O=O+1:V=V+1
3090 NEXT
3100 POKE1,55:POKE56334,1
3110 GOTO3020
```



Stack Pointer

Para conocer cuál es el estado del Puntero del Stack, basta con escribir: Poke2,96:sys2:Print Peek (783)

El Poke coloca un RTS, el SYS lo ejecuta y la Posición 783 contiene el valor del Puntero.

E.H. RUN EEUU.

Gráficas VIC

Víctor Echevarría Ecenarro
C/Egaña, 5, 3º derecha
48010 Bilbao

VIC-20

El primero de los programas permite representar gráficas de funciones en alta resolución sin el empleo del superexpander ni ningún otro cartucho. El programa en sí es corto, pero se ha de copiar con mucho cuidado, porque la base de las gráficas en alta resolución está en la generación de 12×21=252 caracteres, lo cual ocupa de por sí 2,5 Kb de memoria, y el cuerpo del programa ha de entrar en los escasos 1024 bytes restantes en la memoria del VIC standard. Por tanto, cualquier añadido o error al teclearlo puede dar como resultado el fatídico OUT OF MEMORY, con los consiguientes perjuicios.

El manejo del programa es sencillo. Una vez en memoria, se teclea RUN y aparece en pantalla la presentación y una petición de entrada de la función a representar. Se escribe la función y el programa automáticamente crea una línea nueva (nº 29) en el listado, definiéndola (DEF FNA(X)). Después se piden las cotas inferior y superior, tanto en X como en Y, entre las cuales se quiere representar la función. Conviene que estos valores sean simétricos en lo posible; es decir, que tengan el mismo valor absoluto. Ej.: -1 y 1; -3.1416 y 3.1416; etc. Si no lo son, no ocurre nada, pero alguna función puede que no se dibuje en su verdadera magnitud, sino un poco mayor o menor.

Una vez todos los datos dentro, la pantalla se llenará de basura, que se irá borrando poco a poco. Pasado un tiempo la pantalla cambia de color, y se empiezan a dibujar los ejes. Concluido esto, puede ocurrir que se oiga un pitido continuo: esto significa que el ordenador no encuentra por el momento valores de la función representables en el intervalo propuesto, pero el programa continuará rastreando. Cuando se encuentren valores representables, su dibujado irá acompañado de un pitido intermitente. Cuando el sonido cese, la representación habrá concluido. Se pulsa la tecla CTRL y ya tenemos el ordenador listo para otra función. Si se desea representar de nuevo la misma función que acabamos de dibujar (cambiando límites por ejemplo), no hace falta teclearla de nuevo: basta responder con un asterisco a la pregunta de entrada de función, y el programa trabajará con la última que se le haya introducido.

El programa, en líneas generales, ofrece una resolución aceptable: 96×168 puntos. Utiliza, una vez ejecutado, unos 3540 ó 3580 bytes de memoria, según la longitud de la función introducida. Para probar su funcionamiento, se pueden ensayar funciones conocidas tales como SIN(X); 1/X; X^2; X; etc.

```
2 GOTO15
3 GOSUB29:N=36869:V=N+6:R=V+4:POKEV+
3,9
4 POKE56,20:POKE52,20:POKEN,253
5 FORI=5120TO7143:POKEI,0:NEXT:POKER
,45:PRINT"[CLR]"
6 FORI=7680TO8185:POKEI,252:NEXT:FOR
L=0TO11:FORM=0TO20:POKE7707+M*22+L,L
*21+M:NEXTM,L
7 GOSUB45:FORX=IN TOS STEPSP:POKEV,2
20:Y=FNA(X)
11 XX=47+INT(X/SP):YY=83-INT(Y/SY)
12 GOSUB36:POKEV,0:NEXT:WAIT653,4
14 POKEN,240:POKER,27
15 PRINT"[CLR]":GOSUB50:PRINT"[7SPC]
GRAFICAS"
19 PRINT"[5SPC]DE[SPC]FUNCIONES":GOS
UB50
22 PRINT"[CRSRD]ESCRIBA[SPC]SU[SPC]F
UNCION":PRINT:INPUTF$:IFF$="*"THENRU
N3
24 PRINT"[HOM][WHT][17CRSRD]29D[SHIFTE]
FNA(X)="F$:PRINT"G[SHIFTO]3"
25 POKE214,15:POKE198,2:POKE631,13:P
OKE632,13:END
30 POKE198,0:INPUT"[HOM][12CRSRD][BLU]
X[SPC]INF,SUP";IN,S:INPUT"Y[SPC]INF,
SUP";I2,S2
34 SP=(ABS(IN)+ABS(S))/95:SY=(ABS(I2
)+ABS(S2))/167:RETURN
36 IFYY<0ORYY>167THEN44
37 POKEV,0
38 IFXX<0ORXX>95THEN44
39 H=INT(XX/8)*21+INT(YY/8):O=(YY/8-
INT(YY/8))*8:K=5120+8*H+O:P=7-(XX-(I
NT(XX/8)*8))
43 POKEK,PEEK(K)OR(2↑P)
44 RETURN
45 FORXX=0TO95:YY=83:GOSUB36:NEXT:FO
RYY=0TO167:XX=47:GOSUB36:NEXT:RETURN

50 FORI=1TO22:PRINT"*";:NEXT:RETURN
```



SYS para el C-64

En el C-64, una línea puede ser borrada usando:
Poke781,línea:sys59903

LINEA es el número de línea que quieres borrar (de 0 a 24). Puede usarse como subrutina, usando un FOR-NEXT para limpiar la pantalla.

Para hacer un scroll arriba de una línea, SYS 59626.

B.G.A. RUN EEUU.

Memoria libre en el C-64

Cuando preguntamos al C-64...: Print fre(0); a veces nos da como respuesta un número negativo. (Esto no quiere decir que el C-64 tenga memoria negativa, sino que el comando FRE ha representado la memoria libre de otra forma).

Para hallar el verdadero valor, haz:
PRINT FRE(0)+2 16

En lugar del cero de FRE(0), puede usarse cualquier otro número (generalmente el 9 está más a mano).

E.H. RUN EEUU.

Bingo

Rafael García Segura
C/Montes de Oca, 9, 2º
Málaga-7

C-64



Os presento un programa muy simple para jugar al BINGO, con él os evitaréis revolver y extraer bolas. Está realizado en un Commodore 64. Cuando se ejecuta (RUN), aparecerá la presentación y las instrucciones, a continuación la pantalla se divide en dos partes por una línea horizontal, debajo de la cual aparecerá escrito JUGADA 1.... NUMERO (uno al azar), que nos indica que en la primera jugada el número que ha sido extraído es el que figura a la derecha de la palabra NUMERO. Si pulsáis una tecla cualquiera se extrae un nuevo número, distinto del anterior, y la jugada se incrementa en una unidad. Asimismo, el número que había salido en primer lugar pasa a ocupar su posición dentro de una matriz de 10 ×9 en la parte superior de la línea horizontal.

Al pulsar de nuevo una tecla se extrae otra "bola" y la anterior ocupa su lugar en la parte superior y así sucesivamente. Cada extracción va acompañada de un pitido de aviso.

Pienso que las REM son lo suficientemente elocuentes, por lo que omitiré una explicación detallada del programa. Es un programa muy simple y abierto a un gran número de variaciones, tanto en lo referente a la presentación como en el sonido y en color. Cada cual que lo modifique a su gusto.

Al realizarlo he evitado incluir caracteres gráficos, de ahí el uso abusivo de los CHR$. Si te quieres ahorrar unas cuantas pulsaciones de teclas he aquí las equivalencias con las claves que usa la revista:

CHR$ (17)	= [CRSRD]
CHR$ (18)	= [RVSON]
CHR$ (19)	= [HOME]
CHR$ (145)	= [CRSRU]
CHR$ (96)	= [SHIFT C]
CHR$ (146)	= [RVSOF]
CHR$ (147)	= [CLR]

```
0 REM*********************
1 REM*                   *
2 REM*     B I N G O     *
3 REM*                   *
4 REM* R. GARCIA SEGURA  *
5 REM*                   *
6 REM*********************
```

```
7 REM
8 REM
9 REM***PRESENTACION E INSTRUCCIONES
***
10 CLR
15 PRINTCHR$(147):PRINT:PRINT:PRINT
20 PRINTTAB(15)CHR$(18)"B[SPC]I[SPC]
N[SPC]G[SPC]O"CHR$(146)
22 GOSUB4000:PRINTCHR$(147):PRINT:PR
INT
24 PRINT"AL[SPC]SALIR[SPC]LA[SPC]PRI
MERA[SPC]JUGADA,[SPC]PULSE[2SPC]CUAL
-QUIER[SPC]TECLA[SPC]PARA[SPC]CONTIN
UAR"
26 GOSUB4000
29 REM***DIBUJO DE LA LINEA HORIZONT
AL Y DE LAS PALABRA 'JUGADA' Y 'NUME
RO'***
30 PRINTCHR$(147)
40 FORJ=0TO8
60 PRINTCHR$(17)
80 NEXTJ
100 FORK=0TO39
120 PRINTCHR$(96);
140 NEXTK
160 PRINTCHR$(17)CHR$(17)TAB(4)"JUGA
DA";:PRINTTAB(24)"NUMERO"
165 REM
170 REM***GENERACION NOS. AL AZAR***

175 REM
180 DIMA(90):J=1
190 IFJ=91THEN3000
200 X=INT(RND(1)*90)+1
220 IFA(X)=1THEN200
240 A(X)=1
350 GOSUB5000
353 REM
354 REM
355 REM***COLOCACION DE LOS NOS.***
356 REM
357 REM
360 PRINTCHR$(145)TAB(12)J;:IFX<10TH
EN375
370 PRINTTAB(32)X:GOTO380
375 PRINTTAB(32)"[SPC]"X
380 GETA$:IFA$=""THEN380
400 J=J+1
420 PRINTCHR$(19)
460 B=INT(X/10):C=X-B*10
465 IFX<10THENPRINTTAB(1+4*(C-1))C:G
OTO580
480 IFC=0THEN2000
500 FORS=1TOB
520 PRINTCHR$(17)
540 NEXTS
560 PRINTTAB(4*(C-1))X
580 PRINTCHR$(19)
600 FORM=0TO10
620 PRINTCHR$(17)
640 NEXTM
660 GOTO190
2000 FORS=1TOB
2020 PRINTCHR$(17)
2040 NEXTS
2060 PRINTCHR$(145)CHR$(145)TAB(36)X
:GOTO580
2997 REM
2998 REM***FINAL***
2999 REM
3000 GETB$:IFB$=""THEN3000
3040 PRINTCHR$(147):PRINTTAB(167)CHR
$(18)"F[SPC]I[SPC]N[3SPC]D[SPC]E[SPC]
L[3SPC]J[SPC]U[SPC]E[SPC]G[SPC]O":PR
INT:PRINT
3060 GOSUB4000
3100 PRINTCHR$(17)"PARA[SPC]VOLVER[SPC]
A[SPC]JUGAR[SPC]...":PRINT
3120 GOSUB4000
3140 PRINT"PULSE[SPC]CUALQUIER[SPC]T
ECLA"
3160 GETC$:IFC$=""THEN3160
3180 GOTO10
3997 REM
3998 REM***RETARDO DE 3 SEGUNDOS***
3999 REM
4000 TI$="000000"
4020 IFTI$<"000003"THEN4020
4040 RETURN
4997 REM
4998 REM***RUTINA DE SONIDO***
4999 REM
5000 S=54272:FORL=0TO24:POKES+L,0:NE
XT
5010 POKES+3,8:POKES+5,17:POKES+6,13
6
5020 POKES+24,15:POKES+4,65
5030 FORT=1TO10:POKES+0,10:POKES+1,6
0:NEXT
5040 POKES+24,0:POKES+4,64:RETURN
```

Truco del almendruco

José Ignacio Alonso González
Villafranca, 20
Burgos

VIC-20



Con este programa, el VIC 20 se viste con chistera y capa, y empuñando la varita mágica nos asombra con sus dotes de adivinación, procedentes de un cursillo acelerado al que se ve sometido a través de las líneas del programa.

El funcionamiento es sencillo: nosotros pensamos un número del 1 al 64, y el ordenador lo adivinará. Por supuesto que hay truco, pero eso ya cae fuera del campo de la programación. Simplemente nótese que respondiendo sí o no (1 ó 0) 6 veces se obtienen 2 elevado a la 6 (64) combinaciones distintas.

La parte más interesante del programa es el algoritmo que se encarga de generar las tablas que aparecen en pantalla.

Enseña este programa a tus amistades y se sorprenderán del resultado.

```
3 REM EL TRUCO DEL ALMENDRUCO
5 PRINT"[CLR]":POKE36878,15
```

```
7 FORI=128TO250:POKE36876,I:FORJ=0TO
10:NEXT:NEXT
8 POKE36876,0
10 PRINT"[4CRSRD][SPC]PIENSA[SPC]UN[SPC]
NUMERO[SPC]DEL[29SPC]1[2SPC]AL[2SPC]
63"
11 PRINT"[4CRSRD]LUEGO[SPC]PULSA[2SPC]
-ESPACIO-":DIMN(31):P=0
13 GETIN$:IFIN$<>"[SPC]"THEN13
20 FORT=0TO5
30 A=2↑T:B=64-A:C=2↑(T+1)
40 GOSUB500:NEXT
50 PRINT"[CLR][6CRSRD][SPC]EL[SPC]NU
MERO[SPC]QUE[SPC]HABIAS[27SPC]PENSAD
O[SPC]ES[SPC]EL:"
60 FORI=217TO207STEP-5:POKE36876,I:F
ORJ=0TO400:NEXT:NEXT:POKE36876,0
70 PRINT"[2CRSRD][9SPC]"P"[6CRSRD]":
FORZ=0TO500:NEXT
80 FORJ=0TO100:POKE36876,231:POKE368
75,236:NEXT:POKE36876,0:POKE36875,0
90 PRINT"[3CRSRD][SPC]ESTE[SPC]ES[SPC]
EL[SPC]TRUCO[SPC]DEL[29SPC][RVSON]AL
MENDRUCO[RVSOFF]"
100 GOTO100
500 X=0
510 FORI=ATOBSTEPC
515 FORJ=ITOI+(C/2)-1
520 N(X)=J:X=X+1:NEXT:NEXT
530 PRINT"[CLR][2CRSRD]"
540 FORI=0TO28STEP4
550 PRINTTAB(1)N(I)TAB(6)N(I+1)TAB(1
1)N(I+2)TAB(16)N(I+3)
560 NEXT:PRINT:PRINT
570 PRINT"ESTA[SPC]AQUI[SPC]TU[SPC]N
UMERO?":PRINT
580 GET IN$:IFIN$="S"THENP=P+N(0):RE
TURN
590 IFIN$="N"THENRETURN
600 GOTO580
```



Programas que ayudan al usuario

Es corriente que cuando uno escribe una rutina en código máquina, no recuerde la dirección del inicio de la misma. Un buen truco para acordarse es salvar el programa con un nombre como "RUTINA XXXX", donde XXXX es el comienzo de la misma. Si además de esto el que va a utilizarla no es un usuario, podemos llamar al programa: "[CRSR D] SYS XXXXX [3 CRSRU]"; de este modo cuando la rutina acabe de ejecutarse bastará con pulsar return para que esta se ejecute.

S.K. RUN EEUU.

¿Pulse una tecla?

En algunos programas que funcionan con joystick, hay ocasiones en que se le indica al jugador que pulse una tecla para continuar. ¿Porqué no el botón del joystick en lugar de una tecla?

Las siguientes líneas hacen que se pare el programa hasta que se pulse el botón del joystick.

WAIT 37137,32,32	VIC-20
WAIT 145,16,16	C-64 Port 1
WAIT 56464,16,16	C-64 Port 2

E.L.H. RUN EEUU.

Juego de números

Manuel Prieto Ugarte
C/Izaga, 5-4º izda.
Irún

C-64



En este juego, el C-64 y el jugador escogen cada uno un número de cuatro cifras distintas entre sí. Cada uno deberá intentar averiguar el número del otro.
El contador de tiempo indica a quién le toca jugar.
Escribiremos un número y la máquina contestará los muertos + y los heridos @.
Son muertos las cifras acertadas que coincidan en la posición y heridos el resto de las cifras acertadas.
Por ejemplo, si el C-64 tiene el número 8056 (evidentemente no lo sabemos) y escribimos 5206 habrá +@@ el 6 muerto y el 5 y 0 heridos, tampoco conocemos las cifras acertadas, sólo el número de cifras acertadas y cómo.

Cuando sea el C-64 el que juegue y dé un número, debemos escribir tantas "+" como muertos y "@" como heridos haya.
El C-64 a partir de la cuarta tirada empieza a ir despacio.
Por lo general el C-64 siempre acierta el número en la 5 ó 6 tirada y con un tiempo total máximo menor de 20 minutos.

Breve descripción del programa:

10- 80	Programa principal.
100- 190	Inicialización y dibujo del cuadro.
200- 260	Elección de jugador que empieza.
300- 310	Visualización de tiempo.
400- 590	Esperar jugada y dar muertos y heridos.
600- 690	Juega el C-64.
700- 770	Comprobación de número posible o posibilidad de trampas por parte del jugador.
800- 860	Búsqueda de otro número posible.
900- 950	Mensajes de final de juego.
1000-1090	Lectura de muertos y heridos.

```
10 REM JUEGO DE NUMEROS
15 DIM X$(4),X1$(10,4),XM$(4),MU(10)
,HE(10)
20 GOSUB 100
30 GOSUB 200
40 IFJUG=1THENGOSUB400:GOSUB 900
50 GOSUB 600:GOSUB 900
60 GOTO 40
80 END
100 REM INICIALIZACION Y DIBUJO DEL
CUADRO
```

```
110 PRINT "[CLR][2CRSRD][COMMA]";:FO
RI=1TO38:PRINT"[SHIFTC]";:NEXT:PRINT
"[COMMS]";:FORI=1TO20:PRINT"[SHIFTB]
";NEXT
120 PRINT"[COMMZ]";:FORI=1TO38:PRINT
"[SHIFTC]";:NEXT:PRINT"[COMMX]";:FOR
I=1TO20:PRINT"[CRSRL][CRSRU][SHIFTB]
";:NEXT
130 PRINT"[COMMQ]";:FORI=1TO38:PRINT
"[SHIFTC]";:NEXT:PRINT"[COMMW][CRSRU]
";SPC(20);"[COMMR]";
140 FORI=1TO18:PRINT"[CRSRD][CRSRL][SHIFTB]
";:NEXT:PRINT"[CRSRD][CRSRL][COMME][HOM]
[6CRSRD][COMMQ][19SHIFTC][SHIFT+]";
150 PRINT"[18SHIFTC][COMMW][HOM][2CRSRD]
":PRINTSPC(12);"[CRSRU]JUEGO[SPC]DE[SPC]
NUMEROS"
160 PRINT"[CRSRD][2CRSRR]MAQUINA[4CRSRR]
000000[SPC][CRSRR][SPC]JUGADOR[3SPC]
000000"
170 TR=0:TJUG$="000000":TMAQ$="00000
0":XM$=""
180 FORI=0TO3:XM$(I)=CHR$(48+INT(10*
RND(1))):NEXT
185 FORI=0TO2:FORJ=I+1TO3:IFXM$(I)=X
M$(J)THEN180
190 NEXT:NEXT:RETURN
200 REM ELECCION DE JUGADOR QUE EMPI
EZA
210 JUG=-1:I=RND(1):IFI<.5THEN JUG=1

220 EMPEZO=JUG
230 FORI=0TO9:A$(I)=CHR$(48+I):NEXT
240 FORI=1TO5:A=INT(10*RND(1)):B=INT
(10*RND(2)):B$=A$(A):A$(A)=A$(B):A$(
B)=B$
250 NEXT:E1=0:E2=1:E3=2:E4=3
260 RETURN
300 REM CONTABILIZAR E IMPRIMIR TIEM
POS
310 PRINT"[HOM][5CRSRD]";SPC(13+9.5*
(1+JUG));TI$:RETURN
400 REM ESPERAR JUGADA
410 TI$=TJUG$:IFEMPEZO=1THENTR=TR+1
415 PRINT "[HOM][7SPC]TIRADA[SPC]";T
R;"[44SPC]";
420 NU=0:X$="":PRINT"[HOM][6CRSRD][CRSRR]
";:FORI=1TOTR:PRINT"[CRSRD]";:NEXT:P
RINTSPC(21);"[4SPC]";
430 GETX$(NU):GOSUB 300 :IFX$(NU)<"0
"ORX$(NU)>"9"THEN430
440 FORI=0TONU-1:IFNU<>0ANDX$(NU)=X$
(I)THENGOTO420
450 NEXT
455 PRINT"[HOM][6CRSRD][CRSRR]";FOR
I=1TOTR:PRINT"[CRSRD]";:NEXT:PRINTSP
C(21+NU);X$(NU);
457 IFNU=0THENPRINT"[3SPC]";
460 IFNU<3THENNU=NU+1:GOTO430
465 PRINT "[5SPC]";:M=0:H=0
470 FOR I=0TO3:FORJ=0TO3
480 IFX$(I)=XM$(J)ANDI<>JTHENH=H+1
490 IFX$(I)=XM$(J)ANDI=JTHENM=M+1
500 NEXTJ,I
505 M(TR)=M:HE=H
510 IFM<>0THENPRINT"+";:M=M-1:GOTO 5
10
520 IFH<>0THENPRINT"[SHIFTQ]";:H=H-1
:GOTO 520
530 JUG=-JUG:TJUG$=TI$
590 RETURN
600 REM JUEGA LA MAQUINA
610 TI$=TMAQ$:BUSQUEDA=0:IFEMPEZO=-1
THENTR=TR+1
612 PRINT "[HOM][7SPC]TIRADA[SPC]";T
R;"[44SPC]";
615 IF MU(TR-1)=0THENE1=E1+1:E2=0:E3
=1:E4=2
616 IF MU(TR-1)=1THENE2=E2+1:E3=0:E4
=1
617 IF MU(TR-1)=2THENE3=E3+1:E4=0
619 IFTR=1THENGOSUB700:GOTO670
620 GOSUB700:GOSUB800:IFBU=0THEN620
670 PRINT "[HOM][SPC]INDICAR[SPC]MUE
RTOS[SPC]Y[SPC]HERIDOS[5SPC]MUERTOS=
+[SPC],[SPC]HERIDOS[SPC]=[SPC][SHIFTQ]
"
680 PRINT"[HOM][6CRSRD][2CRSRR]";:FO
RI=1TOTR:PRINT"[CRSRD]";:NEXT:PRINTX
$(0);X$(1);X$(2);X$(3);
690 GOSUB 1000:JUG=-JUG:TMAQ$=TI$:RE
TURN
700 E4=E4+1:IFE4>9THENE4=0:E3=E3+1
710 IFE3>9THENE3=0:E2=E2+1
720 IFE2>9THENE2=0:E1=E1+1
730 IFE1>9ORE1>8ANDE2>7ANDE3>6ANDE4>
5THENTR=-1:GOTO900
740 IFE1=E2ORE1=E3ORE1=E4ORE2=E3ORE2
=E4ORE3=E4THEN700
750 X$(0)=A$(E1):X$(1)=A$(E2):X$(2)=
A$(E3):X$(3)=A$(E4):GOSUB300
760 PRINT"[HOM][6CRSRD][2CRSRR]";:FO
RI=1TOTR:PRINT"[CRSRD]";:NEXT:PRINTX
$(0);X$(1);X$(2);X$(3);
770 RETURN
800 FORK1=1TOTR-1
802 IFHE(K1)<>0ORX$(0)<>X1$(K1,1)AND
X$(0)<>X1$(K1,2)ANDX$(0)<>X1$(K1,3)T
HEN804
803 E1=E1+1:E2=0:E3=1:E4=2:RETURN
804 IFHE(K1)<>0ORX$(1)<>X1$(K1,0)AND
X$(1)<>X1$(K1,2)ANDX$(1)<>X1$(K1,3)T
HEN807
805 E2=E2+1:E3=0:E4=1:RETURN
807 IFHE(K1)<>0ORX$(2)<>X1$(K1,0)AND
X$(2)<>X1$(K1,1)ANDX$(2)<>X1$(K1,3)T
HEN811
808 E3=E3+1:E4=0:RETURN
811 IFX$(1)=X1$(K1,1)ANDMU(K1)=0THEN
E2=E2+1:E3=0:E4=0:RETURN
812 IFX$(2)=X1$(K1,2)ANDMU(K1)=0THEN
E3=E3+1:E4=0:RETURN
813 NEXT
820 FORK1=TR-1TO1STEP-1:M=0:H=0:FORI
=0TO3:FORJ=0TO3
830 IFX$(I)=X1$(K1,J)ANDI<>JTHENH=H+
1
840 IFX$(I)=X1$(K1,J)ANDI=JTHENM=M+1

850 NEXTJ,I:IFMU(K1)<>MORHE(K1)<>HTH
ENRETURN
860 NEXT:BUSQUEDA=1:RETURN
900 REM FINAL DE JUEGO
902 IFTR=-1THENPRINT"[HOM][11SPC]HAC
ES[SPC]TRAMPAS";:INPUT"[CRSRU]";A$:P
RINT"[CLR]":STOP
910 IFM(TR)=4ANDMU(TR)=4THENPRINT"[HOM]
[11SPC]EMPATE[SPC]!!";:INPUTA$:PRINT
"[CLR]";:STOP
920 IFM(TR)+ABS(EM+JU)=6THENPRINTS"[HOM]
[8SPC]HAS[SPC]GANADO":INPUT"[CRSRU]"
A$:PRINT"[CLR]":STOP
930 IFMU(TR)+ABS(EM+JU)<>6THEN RETUR
N
940 PRINT"[HOM][2SPC]HE[SPC]GANADO[SPC]
!![SPC]MI[SPC]NUMERO[SPC]ERA[SPC]"XM
$(0);XM$(1);XM$(2);XM$(3):INPUT"[CRSRU]
";A$
950 PRINT"[CLR]":STOP
1000 REM LECTURA DE MUERTOS Y HERIDO
S
1010 X1$(TR,0)=X$(0):X1$(TR,1)=X$(1)
:X1$(TR,2)=X$(2):X1$(TR,3)=X$(3)
1020 PRINTCHR$(13);SPC(12);"[CRSRU][5SPC]
[5CRSRL]";
1030 MU(TR)=0:HE(TR)=0
1040 GETA$
1050 IFA$="+"THENMU(TR)=MU(TR)+1:PRI
NT"+";
```

```
1060 IFA$="[SHIFTQ]"THENHE(TR)=HE(TR
)+1:PRINT"[SHIFTQ]";
1070 IFA$=CHR$(13)THEN RETURN
1080 IF MU(TR)+HE(TR)>4THEN 1020
1090 GOTO 1040
```

Integrales

Víctor Echevarria Ecenarro
Egaña, 5, 3º dcha.
48010 BILBAO.

VIC-20

Calcula por el método de Simpson integrales definidas entre unos límites introducidos por el usuario. Se pide función, límite inferior y superior, y nº de puntos del enrejado. Esto último es el número de porciones en que el programa dividirá la función para calcular su integral. Cuantas más divisiones, mayor precisión en el resultado, pero también se tarda más tiempo en calcular. El usuario deberá decidir el valor que introduce, dependiendo de sus necesidades de precisión y rapidez, y de la clase de función que se estudie.
Cuando ya se ha calculado todo, el programa imprime el signo integral (cadena C$) con los límites dados y escribe el resultado. También señala el tiempo empleado en la operación. El programa es tan simple que no creo que necesite mayor explicación.

```
10 PRINT"[CLR]"
20 PRINT"[CRSRR]*******************
"
30 PRINT"[CRSRR]*[4CRSRR]CALCULO[SPC]
DE[4CRSRR]*"
40 PRINT"[CRSRR]*[4CRSRR]INTEGRALES[4CRSRR]
*"
50 PRINT"[CRSRR]*[4CRSRR]DEFINIDAS[5CRSRR]
*"
60 PRINT"[CRSRR]*******************
"
70 PRINT
80 PRINT"[SPC]INTRODUZCA[SPC]AQUI[SPC]
LA":PRINT"[SPC]FUNCION[SPC]A[SPC]INT
EGRAR"
90 GOTO2000
95 PRINT"[HOM][13CRSRD]";
100 DEFFNA(X)=X↑2
110 A=0:PRINT"[BLU]";
120 A$="F(X)=":C$="[CRSRR][SHIFTU][CRSRL]
[CRSRD][SHIFT-][CRSRL][CRSRD][SHIFT-]
[CRSRL][CRSRD][SHIFT-][CRSRL][CRSRD]
[SHIFTK]":D$="[22CRSRL]"
130 INPUT"COTA[SPC]INFER";I:WE=LEN(S
TR$(I))-2
140 INPUT"COTA[SPC]SUPER";S
145 INPUT"P.ENREJADO";N
146 L1=TI
147 H=(S-I)/N
150 K1=H/3
160 K2=FNA(I)
165 K3=0:K4=FNA(S)
167 J=1
170 FORK=I+HTOS-H STEP H
175 J=J-2*SGN(J)
180 K3=K3+(3-SGN(J))*FNA(K)
190 NEXT
198 R=K1*(K2+K3+K4)
200 GOSUB1000
204 PRINTC$;I;LEFT$(D$,WE)+"[3CRSRL]
[4CRSRU]"S:PRINT"[2CRSRR][CRSRD]"A$;
R"US[CRSRD]"
247 L2=TI
248 PRINT:PRINT"TIEMPO:";INT((L2-L1)
/60);"SEG":PRINT
250 INPUT"MISMA[SPC]FUNCION";Y$
260 IFY$="S"THEN100
265 INPUT"NUEVA[SPC]FUNCION";L$
266 IFL$="N"THENEND
270 RUN
1000 REM**REDONDEO**
1010 R=INT(R*100000+.5)/100000
1020 RETURN
2000 REM**INTRODUCE FUNCION**
2005 PRINT:INPUT"[SPC]";FF$
2010 PRINT"[HOM][WHT][17CRSRD]100DEF
FNA(X)="FF$
2015 PRINT"GOTO95"
2020 POKE214,15:POKE198,2:POKE631,13
:POKE632,13:END
```



Para usuarios del EASY SCRIPT C-64

Suban el volumen de su monitor y a continuación tecleen: (tecla F1) (CTRL+3). Es verdaderamente curioso.

A.I. RUN EE.UU.

Posición 653

La posición de memoria 653, contiene el estado de las teclas SHIFT, CTRL y COMM (la de abajo a la izquierda).
Si ejecutamos: 1 PRINT PEEK (653): RUN
Podremos ver como:

SHIFT = 1
COMM = 2
CTRL = 4

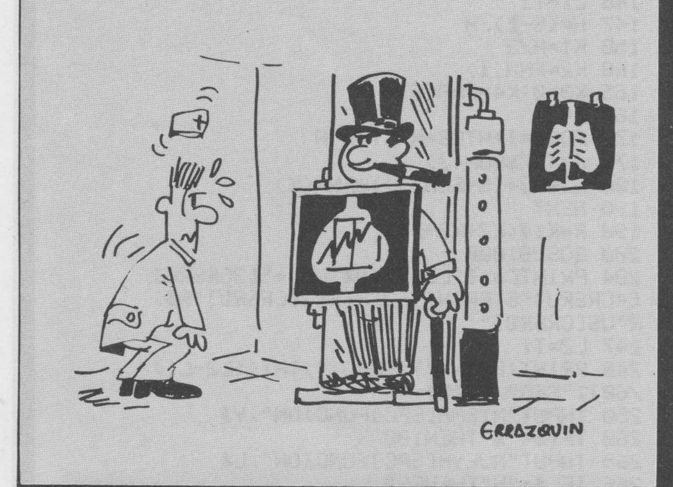
Si pulsamos a la vez SHIFT + COMM, nos da 3 (1+2), las tres teclas a la vez nos da 7 (1+2+4). Esto quiere decir que si en un programa efectuamos la operación PEEK (653) AND 1, nos dirá si SHIFT esta pulsada; AND 2, (COMM) y AND 4 (CTRL). Estas tres teclas son INDEPENDIENTES entre sí y entre las otras del teclado.

A.I. RUN EE.UU.

Gráfico/Gestión

J.M. González Uriarte
C/Umbe, 5-5º c-dcha.
Telf.: (94) 469 05 69
Guecho (Vizcaya)

C-64



En la presentación se pide la gestión anual al dibujar (deberá escribirse esta y pulsar "return"), debiendo dar los datos correspondientes a cada mes. (Para valores mayores que 195 debe buscarse otra escala ya que se sale de la pantalla).
A continuación, se dibujan lentamente en alta resolución los ejes de coordenadas, las ordenadas de cada mes y por último, el gráfico de la gestión anual deseada.
Este programa, en lenguaje máquina, será mucho más rápido.

Comentarios del listado:

1- 167 Presentación. (En 124 se piden datos).
170- 200 Protocolo de entrada en alta resolución.
207- 225 Dibuja el eje vertical.
235- 255 Dibuja el eje horizontal.
265- 300 Dibuja las ordenadas de cada mes.
305- 395 Dibuja las líneas de cada mes. (Si se desea que sea más rápido en este proceso, eliminar de la línea 335 el "STEP.3", con el inconveniente de que saldrá la línea con menos nitidez).
400- 415 Fin.
1000-1600 Bucle para dibujo en alta resolución.
2000-2400 Bucle de presentación.

```
1 REM*****GRAFICOS GESTION********
***
2 REM******VALORES / MES**********
***
3 REM******************************
***
5 PRINT"[CLR]":PRINT:PRINT:PRINT
7 PRINTTAB(12);"[RVSON]GRAFICO[SPC]D
E[SPC]GESTION[RVSOFF]":FOR P=0 TO 10
00:NEXT P
9 PRINT"[CLR]":PRINT:PRINT:PRINT
10 PRINT"INDICAR[SPC]LA[SPC]GESTION[SPC]
A[SPC]EFECTUAR:"
15 PRINT:PRINT:PRINT
20 PRINTTAB(15);"*[SPC]COMPRAS.":PRI
NT
25 PRINTTAB(15);"*[SPC]VENTAS.":PRIN
T
30 PRINTTAB(15);"*[SPC]INVERSIONES."
:PRINT
35 PRINTTAB(15);"*[SPC]BENEFICIOS.":
PRINT
40 PRINTTAB(15);"*[SPC]PERDIDAS.":PR
INT
45 PRINT:PRINT:PRINT:INPUT"*ESCRIBA[SPC]
LA[SPC]GESTION[SPC]QUE[SPC]DESEA[SPC]
REALIZAR";B$
50 PRINT"[CLR]":PRINT:PRINT
55 PRINTTAB(8);"*GRAFICO[SPC]ANUAL[SPC]
DE[SPC]";B$;"*"
56 PRINTTAB(8);"[28COMMU]"
60 PRINT:PRINT:PRINT:PRINT
65 PRINTTAB(8);"PARA[SPC]OBTENER[SPC]
EL[SPC]GRAFICO[SPC]DE[SPC]":PRINT
70 PRINTTAB(5);B$;"[SPC]DEBERAN[SPC]
DAR[SPC]LOS[SPC]VALORES":PRINT
75 PRINTTAB(7);"MENSUALES[SPC]QUE[SPC]
SE[SPC]DEBEN[SPC]DIBUJAR":PRINT
80 PRINTTAB(10);"A[SPC]CONTINUACION.
":PRINT:PRINT
85 FOR W=0 TO 3000:NEXT W
87 DIM S(13),H(13),A$(12)
90 PRINT"[CLR]":PRINTTAB(15);B$:PRIN
TTAB(15);"[9COMMU]"
95 PRINT:PRINT
100 RESTORE
105 FOR I=1 TO 12
110 READ A$(I)
115 NEXT I
117 DATA "ENERO","FEBRERO","MARZO","
ABRIL","MAYO[SPC]","JUNIO","JULIO","
AGOSTO"
120 DATA "SEPTIEMBRE","OCTUBRE","NOV
IEMBRE","DICIEMBRE"
122 FOR I=1 TO 12
124 PRINT"-MES[SPC]DE[SPC]";A$(I);:
INPUT H(I):S(I)=195-H(I)
126 IF H(I)>195 THEN 2000
128 NEXT I
129 PRINT:PRINT:PRINT
130 PRINTTAB(5);"*?DESEA[SPC]CAMBIAR
?....PULSAR[SPC](1)":PRINT
135 PRINTTAB(5);"*?DESEA[SPC]GRAFICO
?....PULSAR[SPC](2)":PRINT
140 GET X$ :IF X$=""THEN 140
145 IF X$="1" THEN 90
150 IF X$="2" THEN 160
155 GO TO 140
160 PRINT"[CLR]":PRINT:PRINT:PRINT:P
RINT
165 PRINTTAB(10);"ESPERE[SPC]UN[SPC]
MOMENTO"
167 FOR W=0 TO 500:NEXT W:PRINT"[CLR]
"
170 BASE=2*4096:POKE53272,PEEK(53272
)OR8
180 POKE53265,PEEK(53265)OR32
190 FOR I=BASE TO BASE+7999:POKE I,0
:NEXT
200 FOR I=1024 TO 2023:POKE I,3:NEXT
205 :
207 REM*****EJE VERTICAL**********
*
210 FOR Y=0 TO 195
215 X=5
220 GOSUB 1000
225 NEXT Y
230 :
235 REM******EJE HORIZONTAL*******
*
240 FOR X=5 TO 313
245 Y=195
250 GOSUB 1000
255 NEXT X
260 :
265 REM******EJES VERTICALES*******
*
```

```
275 FOR X=33 TO 313 STEP 28
280 FOR Y=0 TO 195 STEP 5
290 GOSUB 1000
295 NEXT Y
300 NEXT X
305 REM******* GRAFICO DE MES *****
***
310 DIM X(13)
315 X(1)=5
320 FOR I=1 TO 11
325 X(I+1)=X(I)+28
330 NEXT I
332 FOR I=1 TO 12
335 FOR X=X(I) TO X(I+1) STEP .3
340 R=(X(I)*S(I+1))-(X(I+1)*S(I))
345 T=S(I+1)-S(I)
350 U=X(I)-X(I+1)
355 Y=INT((R-T*X)/U)
360 GOSUB 1000
390 NEXT X
395 NEXT I
400 POKE 1024,16
410 GOTO410
415 END
1000 RO=INT(Y/8)
1100 CH=INT(X/8)
1200 LN=Y AND 7
1300 BI=7-(X AND 7)
1400 BY=BASE+RO*320+CH*8+LN
1500 POKE BY,PEEK(BY) OR (2↑BI)
1600 RETURN
2000 PRINT"[CLR]":PRINT:PRINT:PRINT:
PRINT:PRINT:PRINT
2100 PRINTTAB(8);"PONER[SPC]UN[SPC]V
ALOR[SPC]MENOR[SPC]QUE":PRINT
2200 PRINTTAB(8);"195[SPC]YA[SPC]QUE
[SPC]EL[SPC]GRAFICO[SPC]SE":PRINT
2300 PRINTTAB(8);"SALDRA[SPC]DE[SPC]
LA[SPC]PANTALLA."
2400 FOR K=0 TO 1500:NEXT K:GO TO 90
```



Magia

Sobre los "DATAS"

Si tenemos un programa que contiene unos datas alfanuméricos como por ejemplo: 900 DATA HOLA,A,B,C,ADIOS puede suceder que nos equivoquemos al efectuar el READ, y digamos: 10 READ A

Como A es una variable numérica y "HOLA" es alfanumérica, nuestro ordenador nos dara un error, pero no el "TYPE MISMATCH", sino SYNTAX ERROR en la línea de los DATAS.

Si hacemos el READ en alfanumérico (A$), y los datas son numéricos NO nos dara error, sino que tomará los datas numéricos como alfanuméricos.

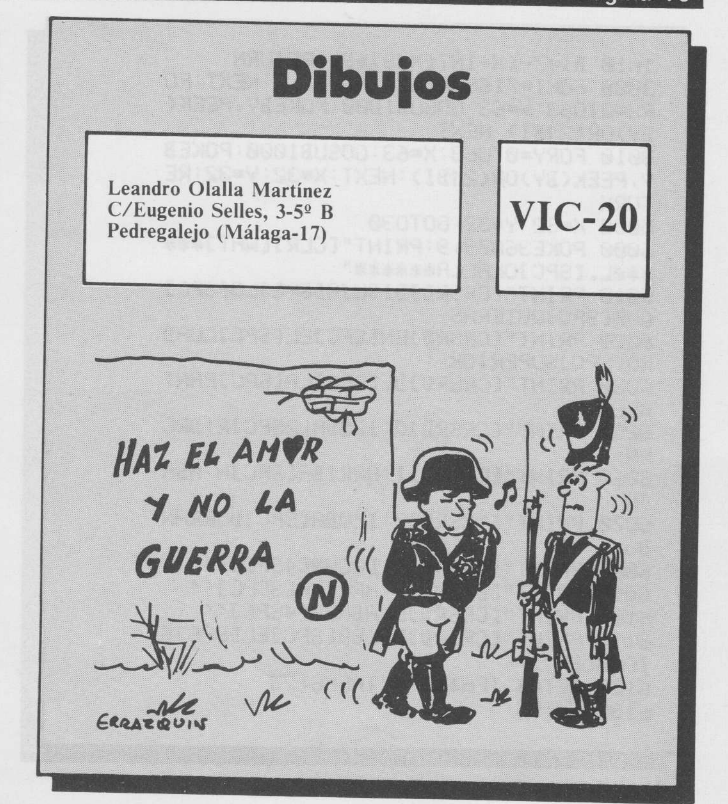
A.I. RUN EE.UU.

Prevención de desastres en la unidad de discos

Si al listar el directorio de un disco te encuentras con un "*" a la derecha del nombre de un programa o de un fichero, no utilices la orden SCRATCH para borrar o podrías causar un auténtico desastre. Utiliza la orden VALIDATE, que realmente borrará el fichero incorrecto.

Si estas con el programa WEDGE, @V se encargará de hacerlo. En caso contrario basta con teclear OPEN15,8,15,"V":CLOSE15

R.A.A. RUN EE.UU.



Dibujos

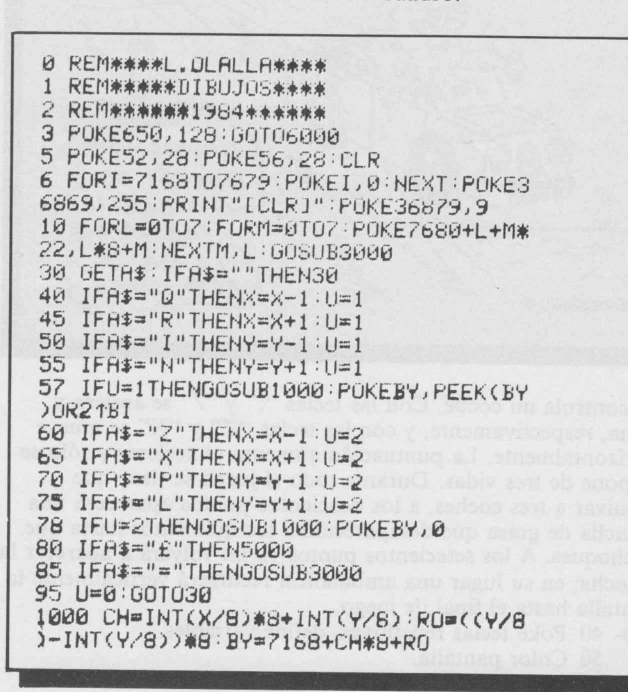
Leandro Olalla Martínez
C/Eugenio Selles, 3-5º B
Pedregalejo (Málaga-17)

VIC-20

Este programa es para dibujar en alta resolución en una porción de la pantalla, para un VIC-20 sin ampliar:

5- 10 Se limpian las direcciones de memoria que van a ser utilizadas y se prepara un cuadro en la parte superior izquierda de la pantalla.
30- 95 Se lleva a cabo el movimiento del punto de dibujo para marcar o borrar.
1000-1010 Son las fórmulas para determinar el byte que va a ser estimulado.
3000-3010 Borra la pantalla, forma el marco y vuelve el punto de dibujo al centro del cuadro.
5000 Coloca el punto en el centro del cuadro.
6000-6130 Comienzo e instrucciones del juego.

Por último, cuando pulséis el signo "=" se borrará la pantalla y se podrá comenzar de nuevo; si se pulsa la "£" el punto de dibujo se colocará en el centro del cuadro.



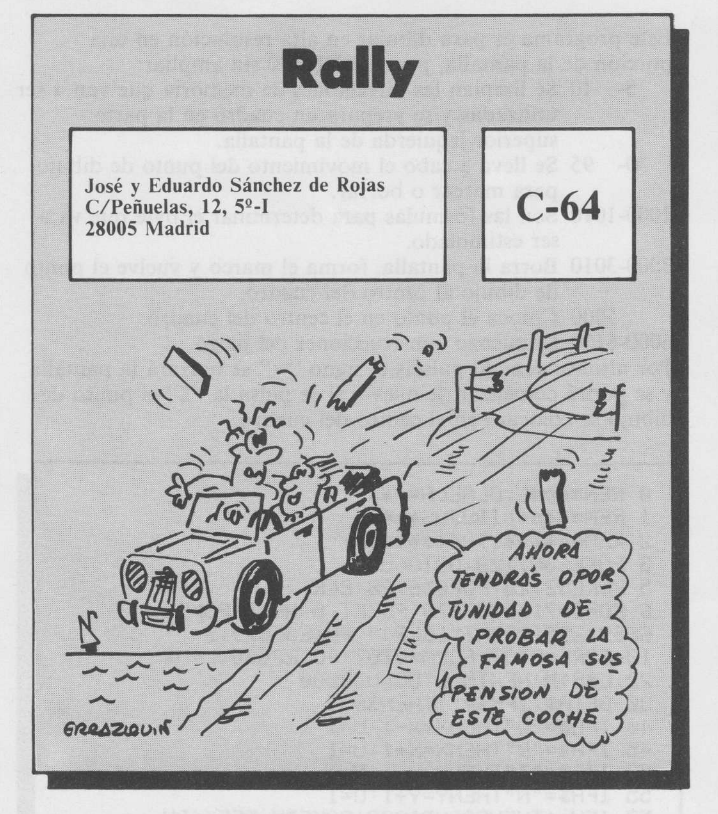
```
0 REM****L.OLALLA****
1 REM*****DIBUJOS****
2 REM******1984******
3 POKE650,128:GOTO6000
5 POKE52,28:POKE56,28:CLR
6 FORI=7168TO7679:POKEI,0:NEXT:POKE3
6869,255:PRINT"[CLR]":POKE36879,9
10 FORL=0TO7:FORM=0TO7:POKE7680+L+M*
22,L*8+M:NEXTM,L:GOSUB3000
30 GETA$:IFA$=""THEN30
40 IFA$="Q"THENX=X-1:U=1
45 IFA$="R"THENX=X+1:U=1
50 IFA$="I"THENY=Y-1:U=1
55 IFA$="N"THENY=Y+1:U=1
57 IFU=1THENGOSUB1000:POKEBY,PEEK(BY
)OR2↑BI
60 IFA$="Z"THENX=X-1:U=2
65 IFA$="X"THENX=X+1:U=2
70 IFA$="P"THENY=Y-1:U=2
75 IFA$="L"THENY=Y+1:U=2
78 IFU=2THENGOSUB1000:POKEBY,0
80 IFA$="£"THEN5000
85 IFA$="="THENGOSUB3000
95 U=0:GOTO30
1000 CH=INT(X/8)*8+INT(Y/8):RO=((Y/8
)-INT(Y/8))*8:BY=7168+CH*8+RO
```

```
1010 BI=7-(X-INT(X/8)*8):RETURN
3000 FORI=7168TO7679:POKEI,0:NEXT:FO
RX=0TO63:Y=63:GOSUB1000:POKEBY,PEEK(
BY)OR(2↑BI):NEXT
3010 FORY=0TO63:X=63:GOSUB1000:POKEB
Y,PEEK(BY)OR(2↑BI):NEXT:X=32:Y=32:RE
TURN
5000 X=32:Y=32:GOTO30
6000 POKE36879,9:PRINT"[CLR][WHT]***
***L.[SPC]OLALLA******"
6010 PRINT"[CRSRD]DIBUJA[SPC]LO[SPC]
QUE[SPC]QUIERAS
6020 PRINT"[CRSRD]EN[SPC]EL[SPC]CUAD
RO[SPC]SUPERIOR
6030 PRINT"[CRSRD]DE[SPC]LA[SPC]PANT
ALLA:
6050 PRINT"[CRSRD]Q:IZQDA[2SPC]R:DRC
HA
6060 PRINT"[CRSRD]I:ARRIBA[SPC]N:ABA
JO
6070 PRINT"[CRSRD]Z:IZQDA[SPC]BORRAN
DO
6080 PRINT"[CRSRD]X:DRCHA[4SPC]''
6090 PRINT"[CRSRD]P:ARRIBA[3SPC]''
6100 PRINT"[CRSRD]L:ABAJO[4SPC]''
6110 PRINT"[CRSRD]PULSA[SPC]EL[SPC]S
IGNO[SPC]=
6120 GETA$:IFA$<>"="THEN6120
6130 GOTO5
```

Rally

José y Eduardo Sánchez de Rojas
C/Peñuelas, 12, 5º-I
28005 Madrid

C-64



Se controla un coche. Con las teclas ";" y "/" se acelera y frena, respectivamente, y con las teclas "Z" y "X" se mueve horizontalmente. La puntuación aumenta al acelerar y sólo se dispone de tres vidas. Durante todo el juego se tiene que esquivar a tres coches, a los doscientos puntos aparecerá una mancha de grasa que desaparecerá a los quinientos hasta que te choques. A los setecientos puntos ya no volverá a aparecer la mancha, en su lugar una ambulancia recorrerá verticalmente la pantalla hasta el final de juego.

10- 40 Poke teclas repetitivas, define variables.
50 Color pantalla.
60- 80 Imprime carretera.
90 Comienza el tiempo.
100-160 Define sprites.
170-200 Control del coche.
210-230 Imprime vidas, tiempo y puntos.
240-260 Comprueba puntos para salida mancha o ambulancia.
290-310 Movimiento del coche.
320-340 Movimiento coche 1.
350-370 Movimiento coche 2.
380-410 Movimiento coche 3.
430-470 Rutina choque.
480-500 Pregunta por nuevo juego e imprime puntuación final.
510-520 Fin
530-580 Movimiento de la mancha o de la ambulancia.
590-640 Datas sprites.

```
0 REM*****************
1 REM* J,E SANCHEZ *
2 REM*    RALLY    *
3 REM*****************
10 POKE650,255:AM=0
30 PRINT"[CLR]":X=171:Y=215:VD=3:S=2
27:VC=150:PC=0
40 M1=54296:M2=54276:M3=54277:M4=542
73:M5=54272
50 POKE53280,0:POKE53281,0
60 FORC=0TO24:POKE1035+40*C,102:POKE
1053+40*C,102:POKE55307+40*C,1
80 POKE55325+40*C,1:NEXT
90 TI$="000000"
100 V=53248
110 POKEV+21,31
120 FORW=2040TO2043:POKEW,13:NEXT:PO
KE2044,198
130 POKEV+39,11:POKEV+43,1:POKEV+23,
16
140 FORN=0TO62:READQ:POKE832+N,Q:NEX
T
150 FORN=0TO62:READZ:POKE12672+N,Z:N
EXT
160 FORN=0TO62:READAM:POKE12288+N,AM
:NEXT
170 GETA$:IFA$="X"THENX=X+5
180 IFA$="Z"THENX=X-5
190 IFA$=";"THENY=Y-5:E=E+10:T=T+10:
S=S+2:PC=PC+10
200 IFA$="/"THENY=Y+5:E=E-5:T=T-5:S=
S-10
210 PRINT"[HOM][CRSRD][2CRSRR][WHT][RVSON]
VIDAS=[RVSOFF]"VD
220 PRINT"[HOM][6CRSRD][2CRSRR][RVSON]
TIEMPO:[RVSOFF]":PRINT"[HOM][8CRSRD]
[2CRSRR]"TI$
230 PRINT"[HOM][CRSRD]"TAB(30)"PUNTO
S:[CRSRD][7CRSRL]"PC
240 IFPC>200ANDPC<400THENPOKEV+43,9:
GOSUB530
250 IFPC>400ANDAM<>0THENGOSUB530
260 IFPC>700THENPOKE2044,192:POKEV+4
3,1:GOSUB530
290 IFY>250THENY=250
300 IFY<0THENY=0
310 POKEV+2,X:POKEV+3,Y:IFX<115ORX>2
30THEN430
320 T=T+20:IFT<0ORT>250 THENT=0
330 Z=INT(RND(1)*40)+140:POKEV+4,Z:P
OKEV+5,T
340 IFX>Z-20ANDX<Z+20ANDY>T-20ANDY<T
+20THEN430
350 E=E+30:IFE>250ORE<0THENE=0
360 POKEV+6,220:POKEV+7,E
370 IFX>200ANDX<240ANDY>E-20ANDY<E+2
0THEN430
380 S=S-6
390 IFS<16THENS=227
400 POKEV+0,120:POKEV+1,S
410 IFX>100ANDX<140ANDY>S-20ANDY<S+2
0THEN430
```

```
420 GOTO170
430 FORMU=15TO0STEP-1:POKEM1,MU:POKE
M2,129:POKEM3,15:POKEM4,40:POKEM5,20
0:NEXT
440 POKEM2,0:POKEM3,0
450 X=171:Y=215:U=U+5:I=I+5:E=0:T=0:
S=250:AM=1
460 VD=VD-1:IFVD=0THEN480
470 FORH=0TO2000:NEXT:GOTO170
480 PRINT"[CLR]":POKE53281,12:POKE53
280,12
490 PRINT"[CRSRD]HAS[SPC]CONSEGUIDO[SPC]
";PC;"PUNTOS"
495 INPUT"[HOM][3CRSRD][RVSON]QUIERE
S[SPC]JUGAR[SPC]DE[SPC]NUEVO[RVSOFF]
(S/N)";A$
500 IFA$="S"THENRESTORE:POKEV+8,50:P
OKEV+9,190:GOTO10
510 PRINT"[CLR][10CRSRD][4CRSRR][BLK]
[RVSON][SPC]F[SPC]I[SPC]N[SPC][RVSOFF]
"
520 END
530 AM=AM+5:IFAM>250THENAM=0
540 POKEM1,15:POKEM3,64:POKE54278,9:
POKEM4,17:POKEM5,37
550 POKEM2,17:POKEM2,16
560 POKEV+8,195:POKEV+9,AM
570 IFX>175ANDX<215ANDY>AM-40ANDY<AM
+40THEN430
580 RETURN
590 DATA7,255,224,7,153,224,113,255,
142,115,255,206,127,195,254,115,189
600 DATA206,115,126,206,3,102,192,3,
37,192,243,66,207,183,126,237,223,12
6
610 DATA251,191,189,253,215,219,235,
243,231,207,1,255,128,0,255,0,7,165
620 DATA224,7,38,224,7,255,224,0,129
,0
630 DATA62,17,224,127,59,243,255,151
,191,251,231,159,241,247,222,112,255
,252
640 DATA59,255,248,127,255,248,127,2
47,252,63,251,222,31,221,191,15,238,
127,30
650 DATA119,254,61,187,254,123,223,6
0,253,238,62,254,222,63,255,63,127,2
55
660 DATA255,254,127,127,156,30,30,0
665 DATA63,255,252,111,231,246,127,2
31,254,111,129,246,255,129
670 DATA255,255,231,255,127,255,254,
127,255,254,127,255,254,127,231,254
675 DATA127,231,254
680 DATA127,255,254,255,255,255,240,
0,15,240,7,207,120,1,30,127,255,254,
63
690 DATA255,252,57,255,156,25,255,15
2,15,255,240
```



Tratamiento de cadenas

Cuando queremos tomar todos los caracteres de una cadena (m$) excepto el primero, podemos escribir:

A$=RIGHT$(M$, len(m$)-1)

Pero es más corto y produce el mismo resultado:

A$=MID$(M$,2)

A.I. RUN EE.UU.

Khan-II

José Antonio Rodríguez Barragán
Rep. Argentina, nº 62-9º 4º
Cornellá (Barna)

VIC-20



Te encuentras en el terrible planeta Khan-II, donde todos tus compañeros, que yacen ahora bajo tierra, fueron víctimas de un ataque masivo de meteoros, sólo tu tremenda habilidad consiguió salvarte del desastre, pero los peligros no hacían más que comenzar. Provisto del más poderoso campo de fuerza, hasta ahora creado, tendrás que activarlo en el momento oportuno para detener a las terribles bombas que te tiran desde los cielos las naves más sofisticadas del imperio. Evita que las bombas den en los cuerpos de tus compañeros y proclámate como el único hombre que ha vencido al imperio.

El programa se maneja desde el teclado con las siguientes teclas.

Pulsar A, movimiento hacia la izquierda.
Pulsar D, movimiento hacia la derecha.
Pulsar ↑, campo de fuerza.
Pulsaciones opcionales.
(Gráficos)
Pulsar F1 (Versión explicada): se encuentran bajo tierra tus compañeros.
Pulsar F3, tus compañeros se transforman en bases enemigas.
Explicación técnica: Sólo las variables más importantes.
Variables.
Y: Define número de hombres.
OL (T): Coloca a tus compañeros al azar.
LO: Número de estrellas.
CK: Marcador.
Cq: Mueve el disparo.
G: Movimiento de la nave.
E1: Indica el estado de la casa.

Nota: Variables como ip o in corresponden a la mecánica del programa.

Cómo hacer funcionar el programa:

Este programa consta de dos partes: la primera crea los gráficos y la segunda es el juego propiamente dicho. Hay que introducirle al ordenador la primera parte (Listado 1), la grabamos por si los fallos y la comprobamos hasta que veamos que funciona correctamente. Hecho esto tecleamos la segunda parte (Listado II) y también la grabamos y comprobamos. Cada vez que queramos jugar, hacemos Load Run a la primera parte y el ordenador se encarga del resto.

Listado I

```
0 PRINT"[CLR]":POKE36879,8:POKE37150
,2:GOSUB224
10 PRINT"[CLR]":POKE52,58:POKE56,28:
DIMN(7)
20 FORT=7168TO7679:POKET,PEEK(T+2560
0):NEXT
30 POKE36869,255
40 READA:IFA=-1THEN200
50 FORT=0TO7:READB:POKE7168+A*8+T,B:
NEXT
60 GOTO40
61 DATA  33 ,195,153,36,24,24,36,153
,195
62 DATA  35 ,24,24,36,36,60,60,24,24

63 DATA  36 ,24,24,24,24,0,0,0,0
64 DATA  39 ,0,0,0,0,0,0,60,60
65 DATA  40 ,24,36,66,255,129,90,60,
24
66 DATA  41 ,24,36,66,255,129,66,36,
0
67 DATA  42 ,24,26,102,88,24,24,36,6
6
68 DATA  43 ,24,88,102,26,24,24,36,6
6
69 DATA  60 ,127,64,88,91,90,66,66,1
26
70 DATA  62 ,254,2,26,218,90,66,66,1
26
71 DATA  26 ,24,90,102,24,24,126,66,
0
72 DATA  59 ,127,65,85,73,65,93,65,1
27
73 DATA  27 ,0,0,0,24,24,0,0,0
74 DATA 23 ,255,255,0,0,0,0,0,0
76 DATA  2 ,3,3,3,3,3,3,3,3
77 DATA  3 ,192,192,192,192,192,192,
192,192
78 DATA  37 ,24,24,36,219,60,24,0,0
79 DATA 38 ,24,24,36,195,36,66,0,0
80 DATA4,0,24,36,90,36,36,36,102
200 DATA-1
210 GOTO2140
224 POKE36897,130:GOSUB310
225 FORA=130TO38STEP-1
250 POKE36897,A:FORT=1TO150:NEXT:NEX
TA:GOTO1000
310 PRINT"[CYN][4CRSRR][2CRSRD]INSTR
UCCIONES:"
320 PRINT"[3CRSRR][CRSRD]-UTILIZA[SPC]
(↑)[SPC]PARA[2SPC]DEFENDERTE"
330 PRINT"[3CRSRR][CRSRD]-UTILIZA[SPC]
(A)[SPC]Y[SPC](D)[SPC]PARA[SPC]MOVER
TE"
340 POKE646,4:PRINT"[2CRSRD]RESISTE[SPC]
TODO[SPC]EL[SPC]TIEMPO[5SPC]QUE[SPC]
PUEDAS!"
350 POKE646,3:PRINT"[4CRSRD][CRSRR]R
EALIZADO[SPC]POR[SPC]J.A.R."
370 RETURN
1000 POKE36877,180
1010 FORL=1TO10
1020 D=INT(RND(1)*5)*50+50
1030 FORM=3TO15
1040 POKE36878,M
1050 FORN=1TOD:NEXTN:NEXTM
1060 FORM=15TO3STEP-1
1070 POKE36878,M:FORN=1TO10
1080 NEXTN,M,L
1090 POKE36878,0
2000 POKE36877,0:GOTO10
2140 POKE198,7:POKE631,76:POKE632,20
7:POKE633,159:POKE634,13
2150 POKE635,82:POKE636,213:POKE637,
13
```

Listado II

```
0 PRINT"[CLR]":NR=4:POKE36878,15:Y=5
:FORT=1TOY:OL(T)=INT(RND(1)*20)+1:PO
KE8120+OL(T),NR:NEXT:LO=50
1 FORT=1TOLO:W=INT(RND(1)*360)
2 POKE7702+W,46:POKE38422+W,6:NEXTT:
IFIL=5THENIL=0:LO=0:GOTO2010
3 TI$="000000":JA=40:PRINT"[HOM][CYN]
[2SPC]###":POKE8095,60:POKE8096,62:B
U=8:UB=19:POKE650,255
4 FORV=1TO20:POKE8142+V,23:POKE38862
+V,5:POKE8098+V,23:POKE38818+V,5:NEX
T
5 A=8094:F=8072:POKE36878,15:DD=200:
GG=36874:CQ=0:U$="PUNTOS:"
130 IP=0:GOSUB21030:CK=CK+10:GOSUB50
095
138 POKE8120,2:POKE8098,2:POKE8141,3
:POKE8119,3
139 POKE38840,5:POKE38861,5:POKE3881
8,5:POKE38839,5
140 O=INT(RND(1)*300)+100:GOSUB2000
214 GOTO130
2000 POKE7724+G,JA:POKEGG,DD:BB=0:PO
KE38444+G,5
2002 FORT=1TO20:NEXT
2004 POKE7724+G,32:POKEGG,0:G=G+1
2008 IFG>20THENG=0:IL=5:LO=LO+2:GOTO
1
2010 IFCK<1000THENKS=1:Y=5:GOSUB5555
5
2020 IFCK>1000ANDCK<3000THENDI=150:G
G=36876:II=2:Y=7:GOSUB55555
2030 IFCK>3000ANDCK<5000THENDD=200:U
U=3:BU=5:KK=0:Y=10:GOSUB55555
2040 IFCK>5000ANDCK<7500THENDD=250:O
O=4:KK=0:BU=3:Y=10:JA=37:GOSUB55555
2045 IFCK>9999THENRUN
2050 GOSUB3000:RETURN
3000 HH=INT(RND(1)*10)+1
3002 IFHH=KSORHH=IIORHH=UUORHH=OOTHE
N3050
3003 RETURN
3050 LL=5:POKE7724+G,JA+1
3052 POKE(7746+G+22*CQ),36:POKE(3846
6+G+22*CQ),7:POKE36876,180
3070 POKE(7746+G+22*CQ),32:POKE36876
,0:UB=17:CQ=CQ+1
3074 FO=PEEK(7746+G+22*CQ)
3075 IFFO=35ORFO=42ORFO=43THENGOSUB5
0000
3076 IFFO=23THENPOKE8098+G,33:POKE36
877,150:FORT=1TO200:NEXT:POKE36877,0

3077 IFFO=40RFO=59THENJI=5:POKE36877
,200:POKE8120+G,33:FORT=1TO1000:NEXT
:H=5:POKE8120+G,32:GOTO50059
3078 IFFO=60ORFO=62THENEL=EL+1:IFEL=
2THENJK=3:GOTO50080
3079 IFFO=39THENPOKEA+PP-22,33:POKE3
6876,200:FORT=1TO100:NEXT:POKE36876,
0:GOTO3081
3080 GOTO3082
3081 UB=14:CK=CK+O
3082 IP=0:POKEA+PP,32:IFCQ>15THENPOK
E38796+G,0:POKE8076+G,61
3083 IFIP=5THENUB=15:GOSUB50059
3084 IFCQ=UBTHENLL=0:CQ=0
21030 IFIN=5THENGOTO21055
21035 GETA$
21040 IFA$="A"THENPP=PP-1:U=43
21050 IFA$="D"THENPP=PP+1:U=42
21051 IFTI$="000130"THENTI$="000000"
:POKE8095,60:POKE8096,62:EL=0
21052 IFA$="[F1]"THENNR=59
21053 IFA$="[F3]"THENNR=4
21055 IFA$=""THEN CC=0:U=35
21056 IFPEEK(A+PP)=61THENGOTO49990
```

```
21057 IFPP=-18THENPP=PP+1
21058 IFPP=1THENPP=PP-1
21060 POKEA+PP,U:POKE38814+PP,3:FORT
=1TO25:NEXTT:
21067 IFA$="↑"THENPOKEF+PP,39:POKE36
876,225:FORT=1TO50:NEXT:POKE36876,0:
IN=5:POKE38792+PP,4
21068 KK=KK+1:IFKK=BUTHENPOKEF+PP,32
:KK=0:IN=0:POKE650,255
21069 IFLL=5THEN3050
21070 POKEA+PP,32
21098 RETURN
49990 POKE8116+PP,26:FORT=1TO500:NEX
T:POKE8116+PP,32:POKE8138+PP,33:KU=5

49992 POKE38858+PP,2:GOTO50010
50000 POKEA+PP,33:POKE38814+PP,2
50010 POKE36877,220:FORL=15TO0STEP-1
:POKE36878,L:FORM=1TO300
50020 NEXTM,L:POKE36877,0:POKE36878,
15:IP=5:POKEA+PP,32:POKEF+PP,32:CQ=0
:GOSUB50100
50030 IFKU=5THENPOKE8138+PP,32:KU=0:
GOSUB55555:GOTO3083
50059 VV=7684:JK=JK+1:IFH=5THENPOKEA
+PP,32:POKE36877,0:H=0
50070 IFJK=2THENVV=7683
50080 IFJK=3THENPRINT"[CLR]GAME[SPC]
OVER":END
50090 POKEVV,32:A=8094:PP=0:FORR=1TO
5:POKE36876,200:FORT=1TO30:NEXTT:POK
E36876,0
50091 NEXTR
50092 IFJI=5THENJI=0:GOTO3078
50093 RETURN
50095 PRINT"[YEL][HOM]"TAB(8)U$CK:RE
TURN
50100 FORV=1TO20:POKE8098+V,23:NEXTV
:PRINT"[HOM][18CRSRD][19SPC]":RETURN

55555 FORT=1TOY:POKE8120+OL(T),32:NE
XTT
55557 FORT=1TOY:OL(T)=INT(RND(1)*20)
+1:POKE8120+OL(T),NR:POKE38840+OL(T)
,3:NEXTT:RETURN
```



Cero abreviado

En las sentencias de asignación de variables, como A=0, el cero puede sustituirse por un punto (A=.). Es aproximadamente un 20% más rápido.

RUN EE.UU.

Comillas

El truco para poner comillas en las palabras de un print es:

PRINT CHR$(34);"HOLA";CHR$(34)

J.O. RUN EE.UU.

Datas con cero

Los números cero que se encuentran en sentencias data pueden no ponerse. Por ejemplo.

10 data 20,4,0,2,4,0

Es lo mismo que...

10 data 20,4,,2,4

M.S. RUN EE.UU.

Frontón

Manuel Llovet Viña
Carrer Navarra, nº 17
Horta de Sant Joan (Tarragona)

C-64

Frontón funciona con el joystick, en la posición del 1er jugador o por el teclado en este caso con las teclas del cursor.

Al principio el programa te pregunta si quieres jugar con el joystick o el teclado, tú le respondes pulsando la tecla j o t. Luego te pregunta el nivel de juego 0 el más sencillo al 5 el más complicado, tú pulsas el número del nivel correspondiente. Luego no tienes más que pulsar (RETURN) y ya empiezas a jugar.

Gana la partida el que primero llega a 21 puntos. ¡Buena suerte!

```
1 REM *****FRONTON*****
2 REM ***ESCRITO POR***
3 REM ****M. LLOVET****
50 PRINT"[CLR]":POKE53280,2:POKE5328
1,0:R=0
60 PRINT"JOYSTICK[SPC]O[SPC]TECLADO[SPC]
(J/T)?"
70 GET B$:IF B$<>"J"AND B$<>"T"GOTO
70
80 INPUT"NIVEL[SPC]DE[SPC]DIFICULTAD
?[SPC]ELIJE[SPC]ENTRE[SPC]0[SPC]A[SPC]
5";H
90 IF H=0 THEN R=2
95 IF H<0 OR H>5 THEN80
100 K=0:J=0
110 F$="[YEL][RVSON][SPC][RVSOFF]"
120 PRINT"[CLR]";:FORI=1TO39:PRINTF$
;:NEXT:PRINT"[CRSRU]"
130 FORI=1 TO 23: PRINTF$:NEXTI
140 FORI=1 TO 39: PRINTF$;:NEXTI:PRI
NT"[HOM]"
150 TI$="000000"
160 W=0:L=0:M=0:GOSUB2500
170 PRINT"[23CRSRD][3CRSRR]YO[SPC]=[SPC]
",:PRINTL,:PRINT"[HOM]"
180 PRINT"[HOM]":PRINT"[23CRSRD][14CRSRR]
TU[SPC]=",:PRINTM,:PRINT"[HOM]"
190 E=0
200 X=1:Z=4
210 Y=INT(21*RND(1)+2)
220 DX=1:DY=1:DZ=1
230 POKE1024+X+40*Y,81
```

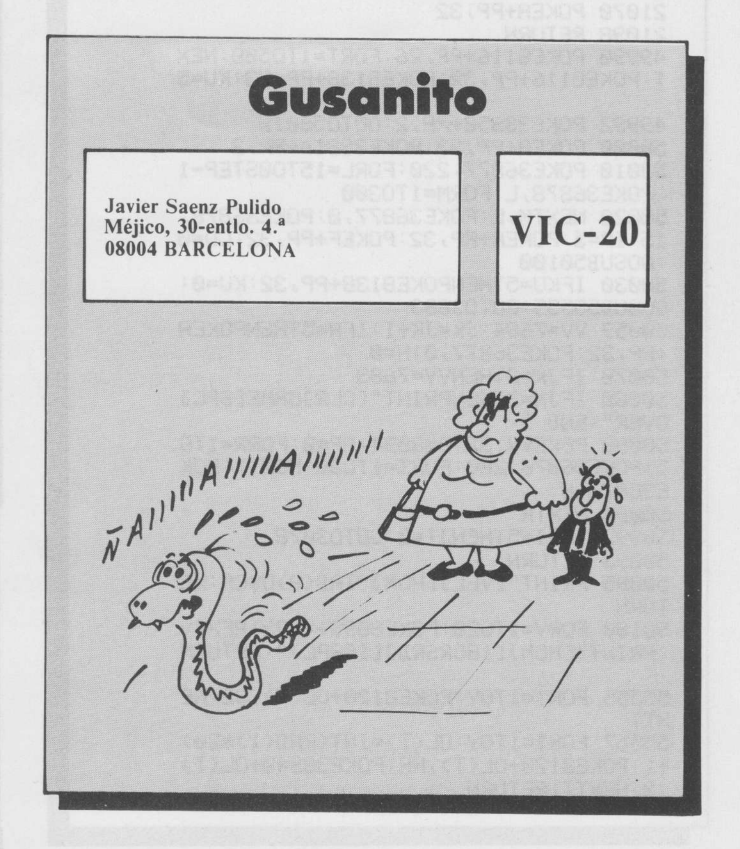
```
240 POKE1024+35+40*Z,97
250 FOR T=1 TO3:NEXT
260 PRINT"[HOM]":PRINT"[23CRSRD][25CRSRR]
",:PRINTTI$;
270 PRINT"[HOM]"
280 POKE1024+X+40*Y,32
290 POKE1024+35+40*Z,32
300 X=X+DX
310 IFX=1 GOTO390
320 IFX=38 GOTO 1000
330 IF X=35 GOTO 350
340 GOTO 360
350 IF Y=Z OR Y=Z+1 OR Y=Z-1 GOTO200
0
360 GOTO400
370 W=0: GOSUB2500
380 Z=4
390 DX=-DX
400 Y=Y+DY
410 IF Y<=1OR Y>=23 THEN DY=-DY
420 Z=Z+DZ
430 IF Z<=2 OR Z>=22 GOTO530
450 IF B$="J"THEN3000
460 GETA$
470 IF A$="[CRSRD]"THEN DZ=-1
480 IF A$="[CRSRU]"THEN DZ=-2
490 IF A$="[CRSRR]"THEN DZ=+1
500 IF A$="[CRSRL]"THEN DZ=2
510 IF A$<>"[CRSRD]"AND A$<>"[CRSRU]
" AND A$<>"[CRSRR]" AND A$ <>"[CRSRL]
" THEN DZ=0
530 DZ=-DZ
540 GOTO 230
590 IF L>=21THEN K=K+1
600 IF M>=21 THEN J=J+1
610 PRINT"[CLR]"
620 PRINT"[2CRSRD]YO[SPC]TENGO",K,"P
ARTIDAS"
630 PRINT"[4CRSRD]TU[SPC]TIENES",J,"
PARTIDAS"
640 PRINT"[6CRSRD]QUIERES[SPC]HACER[SPC]
OTRA[SPC]PARTIDA[SPC](S/N)?"
650 GETV$: IF V$<>"S"ANDV$<>"N"GOTO
650
660 IF V$="N"THEN END
670 PRINT"[CLR]"
680 GOTO 110
1000 GOSUB2500:IF R<>2 GOTO1010
1005 E=E+1:GOSUB1550:IF E<R GOTO 200

1010 L=L+1:GOSUB1550
1020 PRINT"[23CRSRD][3CRSRR]YO[SPC]=
[SPC]",:PRINTL,:PRINT"[HOM]"
1030 IF L<21 GOTO 190
1040 GOTO 590
1510 S=54272:POKES+24,15:POKES+5,100
:POKES+1,25:POKES+0,177:POKES+4,17
1520 FORP= 1 TO 100:NEXT
1530 POKES+4,0:POKES+5,0:POKES+6,0
1540 RETURN
1550 S=54272:POKES+24,15:POKES+5,100
:POKES+1,12:POKES+0,216:POKES+4,17
1560 FORP=1 TO 100:NEXT
1570 POKES+4,0:POKES+5,0:POKES+6,0
1580 RETURN
2000 W=W+1:GOSUB1510:GOSUB2500
2005 IF W<H GOTO 380
2010 M=M+1:GOSUB1510
2020 PRINT"[HOM]":PRINT"[23CRSRD][14CRSRR]
TU[SPC]=",:PRINTM,:PRINT"[HOM]"
2030 IF M<21 GOTO 370
2040 GOTO 590
2500 PRINT"[HOM]",:PRINT"[14CRSRR]",
:PRINTH-W
2510 RETURN
3000 JV=PEEK(56321)AND15:FB=-((PEEK(
56321)AND16)=0)
3001 PRINT"[HOM]",:PRINTJV,:PRINT"[HOM]
"
3010 IF JV=14 THEN DZ=-1
3015 IF JV=11 THEN DZ=-2
3020 IF JV=13 THEN DZ=+1
3025 IF JV=7 THEN DZ=2
3030 IF JV=15 OR JV=5 OR JV=9 OR JV=
10 THENDZ=0
3040 GOTO 230
```

Gusanito

Javier Saenz Pulido
Méjico, 30-entlo. 4.ª
08004 BARCELONA

VIC-20



Es para un VIC-20 sin necesidad de ampliación.
A continuación os doy unas pequeñas explicaciones sobre el programa:

- 1-8: Se pide el nivel de dificultad y se imprime el título.
- 15-30: Se traslada el juego de caracteres y se crea los nuestros.
- 80-130: Se dibuja el marco del juego.
- 135-146: Se imprimen los monstruitos.
- 150: Se inicializan las variables de utilidad.
- 170: Parpadeo de nuestro gusanito.
- 190-290: Programa principal, en el cual según la tecla que hayas pulsado (dentro de las de control) irás a una instrucción determinada.
- 300-340: Rutina para imprimir los objetivos. Se elige al azar si se imprime una manzana o una cereza, en que se dan más oportunidad a las cerezas que dan menos puntuación.
- 400-410: Aquí se produce el movimiento del gusanito. Hay dos variables: la de la cabeza denominada A y la de la cola del gusanito denominada B. La variable A se incrementa según lo que se haya pulsado en el teclado, mientras que la B se incrementa por los valores que va leyendo en el vector V(x). Los valores almacenados en este vector son 22, -22, 1, -1 y vienen determinados por el movimiento que haya efectuado anteriormente el gusanito; es decir si el gusanito se ha desplazado hacia abajo se almacenará 22, si se desplaza hacia la izquierda se almacenará un -1.
- 500-610: Esta rutina es en caso de haber colisionado con algo, el gusanito. (510) esta es en caso de colisionar con el recuadro; (530-540) estas es en caso de comerte una

manzana o cereza respectivamente; (520) en caso de colisionar con el mismo gusanito; (550) esta es en caso de toparte con algún monstruito.
620-640: Aquí te da un punto por cada objetivo que te has comido.
650-690: Final de juego.
700-745: En caso de colisionar con las paredes busca la dirección a seguir.
800-930: DATAS para efectuar el título del comienzo y para la creación de los caracteres.

Después de esta explicación del programa, pasaré a su utilización.
Tu objetivo es dirigir el gusanito por el recuadro con las teclas /A/=izquierda; /D/=derecha; /W/=arriba; /X/=abajo (¿se nota que no poseo joystick?)
Cada 20 objetivos que hayan aparecido te volverá a reducir el gusanito a su primitiva forma, sumándote puntos por la longitud que hayas conseguido lograr.
El truco de la cuestión está en que por lo menos de esos 20 objetivos hayas conseguido uno, de esta manera seguirá la partida, aumentándote la cantidad de monstruitos nocivos para la salud de nuestro gusanito.
En caso de que poseáis mayor capacidad de memoria que la mía (que es la original) podéis mejorar el programa poniendo récord, dos jugadores, tiempo, etc.

Espero vuestras noticias, sobre lo que hayáis conseguido.

```
1 PRINT"[CLR]"
2 READZ1:Z=96+128:IFZ1=0THENGOTO10
3 IFZ1=169THENZ=95+128
4 IFZ1=119THENZ=95
5 IFZ1=143THENZ=95+128
6 IFZ1=120THENZ=100
7 IFZ1=235THENZ=42
8 POKEZ1+7700,Z:POKEZ1+7700+30720,2:
GOTO2
10 PRINT"[17CRSRD]"
12 INPUT"DIFICULTAD[SPC](0/9)";K:IFK
<0ORK>9THENRUN
15 A=7168:FORI=ATOA+511:POKEI,PEEK(I
+25600):NEXT:N=7167
20 READA:IFA<>-1THENN=N+1:POKEN,A:GO
TO20
30 PRINT"[CLR]":POKE56,28:POKE52,28:
POKE36869,255
80 Y=38399:DIMV(20)
81 FORX=7680TO7701
82 Y=Y+1
83 POKEX,4:POKEY,7
84 POKEX+484,4:POKEY+484,7
85 NEXT:Y=38378
100 FORX=7680TO8142STEP22
110 Y=Y+22
120 POKEX,4:POKEY,7
125 POKEX+21,4:POKEY+21,7
130 NEXTX
135 FORX=1TOK*2
140 C=INT(RND(1)*437)+7725
145 IFPEEK(C)<>32GOTO140
146 POKEC,2:POKEC+30720,0:NEXTX
150 A=7703:B=A:C=38423:B$="D":TF=40:
K=0:A1=0:B1=0
170 FORX=1TO8:POKEA,32:FORY=1TO200:N
EXTY:POKEA,5:POKEC,2:FORZ=1TO100:NEX
TZ:NEXTX
175 Y=30720:Z=36878
180 GOSUB300
190 GETA$:IFOB>20THENOB=0:K=K+1:J=9:
GOTO622
191 IFA1>19THENA1=0
192 IFB1>19THENB1=0
200 IFA$=""ORA$=V$THENA$=B$:GOTO210
210 TA=TA+1:IFTA=TFTHENPOKEC,32:GOSU
B300
215 IFA$="X"GOTO230
217 IFA$="A"GOTO260
219 IFA$="D"GOTO280
221 IFA$="W"GOTO225
223 GOTO190
225 A=A-22:B$=A$:A1=A1+1:V(A1)=-22:I
FPEEK(A)<>32THENJ=22:GOTO500
228 GOTO400:GOTO190
230 A=A+22:B$=A$:A1=A1+1:V(A1)=22:IF
PEEK(A)<>32THENJ=-22:GOTO500
250 GOTO400:GOTO190
260 A=A-1:B$=A$:A1=A1+1:V(A1)=-1:IFP
EEK(A)<>32THENJ=1:GOTO500
270 GOTO400:GOTO190
280 A=A+1:B$=A$:A1=A1+1:V(A1)=1:IFPE
EK(A)<>32THENJ=-1:GOTO500
290 GOTO400:GOTO190
300 TA=0:OB=OB+1:IFOB>20THENRETURN
310 D=INT(RND(1)*10)+1
311 IFD>8THEND=1:P=5:GOTO315
312 D=0:P=2
315 C=INT(RND(1)*484)+7703
320 IFPEEK(C)<>32GOTO310
330 POKEC,D:POKEC+Y,P
340 RETURN
400 POKEA,5:POKEA+Y,2:POKEB,32
410 B1=B1+1:B=B+V(B1)
460 GOTO750
500 REMAQUI CUANDO COLISION
510 IFPEEK(A)=4GOTO700
520 IFPEEK(A)=5THENFORX=1TO100:POKEZ
,15:POKEZ-2,215:NEXT:GOTO620
530 IFPEEK(A)=1THENSC=SC+4
540 IFPEEK(A)=0THENSC=SC+2
550 IFPEEK(A)=2THENPOKE36878,15:POKE
36876,200:FORX=1TO500:NEXT:POKE36876
,0:GOTO650
560 POKE36875,130:POKEZ,15:FORX=1TO1
00:NEXT:POKE36875,0
600 POKEA,5:POKEA+Y,2:LO=LO+1
610 GOSUB300:GOTO750
620 A=A+J:POKEZ,0:GOTO650
622 POKE36875,130:POKEZ,15:FORX=15TO
0STEP.25:POKEZ,INT(X):FORY=1TO500:NE
XTY
623 NEXTX:POKE36875,0
624 IFLO=0GOTO650
625 POKEB,32
628 B1=B1+1:IFB1>20THENB1=1
629 B=B+V(B1)
630 POKEZ,15:POKE36875,200:FORM=1TO1
00:NEXTM:POKE36875,0
640 IFB<>ATHENSC=SC+1:GOTO625
642 POKEA,32
644 LO=0
645 IFJ=9THENGOTO135
650 PRINT"[8CRSRD][CRSRR]TUS[SPC]PUN
TOS:"SC
660 PRINT"[CRSRD][CRSRR]F1[SPC]OTRO"
670 GETZ$:IFZ$=""GOTO670
680 IFZ$="[F1]"THENCLR:POKE36869,240
:RUN
690 GOTO670
700 A=A+J:A1=A1-1
710 IFPEEK(A-1)<>4ANDPEEK(A-1)<>5AND
V$<>"A"THENB$="A":GOTO750
720 IFPEEK(A+1)<>4ANDPEEK(A+1)<>5AND
V$<>"D"THENB$="D":GOTO750
730 IFPEEK(A+22)<>4ANDPEEK(A+22)<>5A
NDV$<>"X"THENB$="X":GOTO750
740 IFPEEK(A-22)<>4ANDPEEK(A-22)<>5A
NDV$<>"W"THENB$="W":GOTO750
745 GOTO650
750 IFB$="W"THENV$="X"
760 IFB$="X"THENV$="W"
770 IFB$="A"THENV$="D"
780 IFB$="D"THENV$="A"
790 GOTO190
```

```
800 DATA97,98,99,107,108,109
810 DATA119,120,130
820 DATA136,137,138,143,146,148,152
830 DATA158,163,164,165,168,169,170,
174
840 DATA180,182,190,192
850 DATA202,203,204,207,208,209,212,
214,217,218,219
860 DATA229,231,235,239,241
870 DATA246,248,251,252,253,257,261,
263
880 DATA268,270,273,275,279,283,284,
285
890 DATA290,292,301
900 DATA312,313,314,323,0
910 DATA8,8,28,20,36,231,231,231,2,4
,28,62,62,62,54,20
920 DATA56,84,56,16,108,130,130,68,0
,0,0,0,0,0,0,0
930 DATA255,255,255,255,255,255,255,
255,0,60,66,82,74,66,60,0,-1
```

Memorya

Miguel Bermejo
Moriones, 6-10 Esc. 2° 9º B
50006 ZARAGOZA

C-64

Como su nombre indica, se basa en un conocido juego de niños que se utiliza para desarrollar la memoria visual; lo cual no quiere decir que no puedan pasar sus buenos ratos con él los que no son tan niños...

Consta de un tablero con 9 cartas horizantales por 5 verticales (45 cartas en total) las cuales se encuentran boca abajo. El juego consiste en que en cada turno de ambos jugadores levanten dos cartas, primero una y a continuación la otra tratando de encontrar las 22 parejas.

Cada jugador en su turno levanta 2 cartas, indicando sus coordenadas si al levantar la primera recuerda donde estaba la pareja (si ha aparecido ya), pulsa sus coordenadas y gana 1 punto y continúa jugando. Parece sencillo ¿no?

El programa esta basado en Sprites multicolores, como hay 23 dibujos distintos, lógicamente hay gran cantidad de Datas ¡Cuidado al copiarlo? Aunque si os equivocáis en alguno será fácil verlo al ver los dibujos.

Al comenzar el juego os aparecen las instrucciones, hasta que pulséis una tecla, una vez pulsada os aparecerá el tablero con las cartas cubiertas, pero debéis de esperar aproximadamente 1 minuto para que el C-64 baraje y distribuya las cartas, hasta que debajo del jugador nº 4 aparezca la leyenda "OBJETO 1", y ya podéis empezar a hacer memoria.

Si algún SPRITE no os gusta o tenéis por ahí otros más "monos" ya sabéis, lo cambiáis en los datas, pero recordad que han de ser Sprites multicolor.

La estructura del programa es como sigue:

10 Introduce los Datas en posiciones de memoria.
18-90 Inicializa variables.
100-170 Presenta instrucciones.
180-260 Dibuja tablero.
263 Envía a borrar cartas encontradas.
270 Si es 1ª pasada baraja y reparte las cartas.
275-360 Entrada para Objeto 1º, inicializa sprite y lo coloca.
361-420 Idem para Objeto 2º.
430-530 Analiza si ambas cartas son iguales. Si son distintas lo manda a empezar de nuevo.

RUTINAS
1000-1060 Baraja y reparte cartas (sprites), en el tablero.
1100-1220 Coloca cada sprite en la posición que dan las coordenadas.
2000-2400 Si ambas cartas son iguales busca la música y limpia el lugar ocupado por ambas.
2500-2900 Limpia el tablero de todas las parejas encontradas.
3000-3090 Ganador y posible repetición del juego.
14000-14030 Carga Datas en posiciones de memoria.
15000-16570 Datas de Sprites, Colores de sprites, Músicas.

Si después de todo esto os queda alguna duda no reparéis en escribirme os contestaré a todos.

```
1 REM  MEMORYA POR MIGUEL BERMEJO
10 GOSUB14000
18 CLR
30 DIMX1(22),Y1(22),X2(22),Y2(22)
90 V=53248:J1=0:J2=0:CT=0:S=54272:H=
0
100 PRINT"[CLR]":POKE53280,10:POKE53
281,10:PRINT"[COMM2]"
110 PRINT"[7CRSRR][CRSRD]VAMOS[SPC]A
[SPC]JUGAR[SPC]AL[SPC]MEMORYA"
115 PRINT"[3CRSRR][CRSRD][4SPC]JUEGA
[SPC]EL[SPC]JUGADOR[SPC]AMARILLO"
120 PRINT"[YEL][2CRSRD]1.-[SPC]TIENE
[SPC]QUE[SPC]DAR[SPC]SUS[SPC]COORDEN
ADAS[CRSRR]PARA[2SPC]EL[SPC]PRIMER[SPC]
OBJETO,";
130 PRINT"[CRSRR]PRIMERO[SPC]LA[SPC]
LETRA[SPC]Y[4SPC]LUEGO[SPC]EL[SPC]NU
MERO"
140 PRINT"[BLU][CRSRD]2.-ESPERE[SPC]
QUE[SPC]APAREZCA[SPC]EL[SPC]PRIMER[SPC]
OBJETO"
150 PRINT"[RED][CRSRD]3.-[SPC]REPITA
[SPC]LA[SPC]MISMA[SPC]OPERACION,[CRSRR]
PARA[SPC]EL[SPC][CRSRR]SEGUNDO[SPC]O
BJETO"
155 PRINT"[CYN][CRSRD]4.-SI[SPC]AMBO
S[SPC]SON[SPC]IGUALES[SPC]CONTINUE,[SPC]
SI[SPC]NO[SPC]JUEGA[SPC]EL[SPC]OTRO[SPC]
JUGADOR"
157 PRINT"[BLK][CRSRD]5.-ATENCION[SPC]
HAY[SPC]UN[SPC]OBJETO[SPC]DESPAREJAD
O"
160 PRINT"[COMM4][2CRSRD]SI[SPC]HA[SPC]
ENTENDIDO[SPC]PULSE[SPC]UNA[SPC]TECL
A"
170 GETA$:IFA$=""THEN170
180 POKE53280,8:POKE53281,8:PRINT"[RED]
[CLR][2CRSRL]";
```

Le presentamos a SPINNAKER.
Para que también nuestros hijos aprendan jugando con el ordenador.

Muchos padres ya conocen que el ordenador doméstico puede servir para que nuestros hijos jueguen y pasen largos ratos de ocio. Con SPINNAKER los **juegos de ordenador** además sirven **para aprender**.

SPINNAKER es la firma líder mundial en juegos educativos para ordenador. Ahora llega a nuestro país para que también nuestros hijos aprendan jugando con el ordenador.

Nuestros juegos están llenos de emoción, entretenimiento, color, música... Pero además poseen un alto grado de valor educativo, ayudando a los niños a que desarrollen sus habilidades y aprendan de una divertida manera. Aprovechando al máximo las nuevas posibilidades que nos ofrece el mundo de las computadoras.

En su elaboración han intervenido especialistas en la educación, la didáctica, la música, los gráficos y la programación, que han realizado un cuidado producto especialmente pensado para la educación.

Estos son los motivos por los que miles de niños de todo el mundo son verdaderos fans de SPINNAKER. Y sus padres también.

Si Ud. realmente desea que su ordenador familiar sea una ayuda útil para el futuro de sus hijos, recuerde este nombre: SPINNAKER: Una divertida familia de juegos educativos.



DISPONEMOS DE VERSIONES PARA APPLE II, Commodore 64 y Sinclair Spectrum.

Precio especial de introducción 2.400 pts.



Gran Via Carlos III, 97 K. 08028 BARCELONA



SPINNAKER SOFT COR. CAMBRIDGE, MASS. USA.



¿Utiliza sólo un

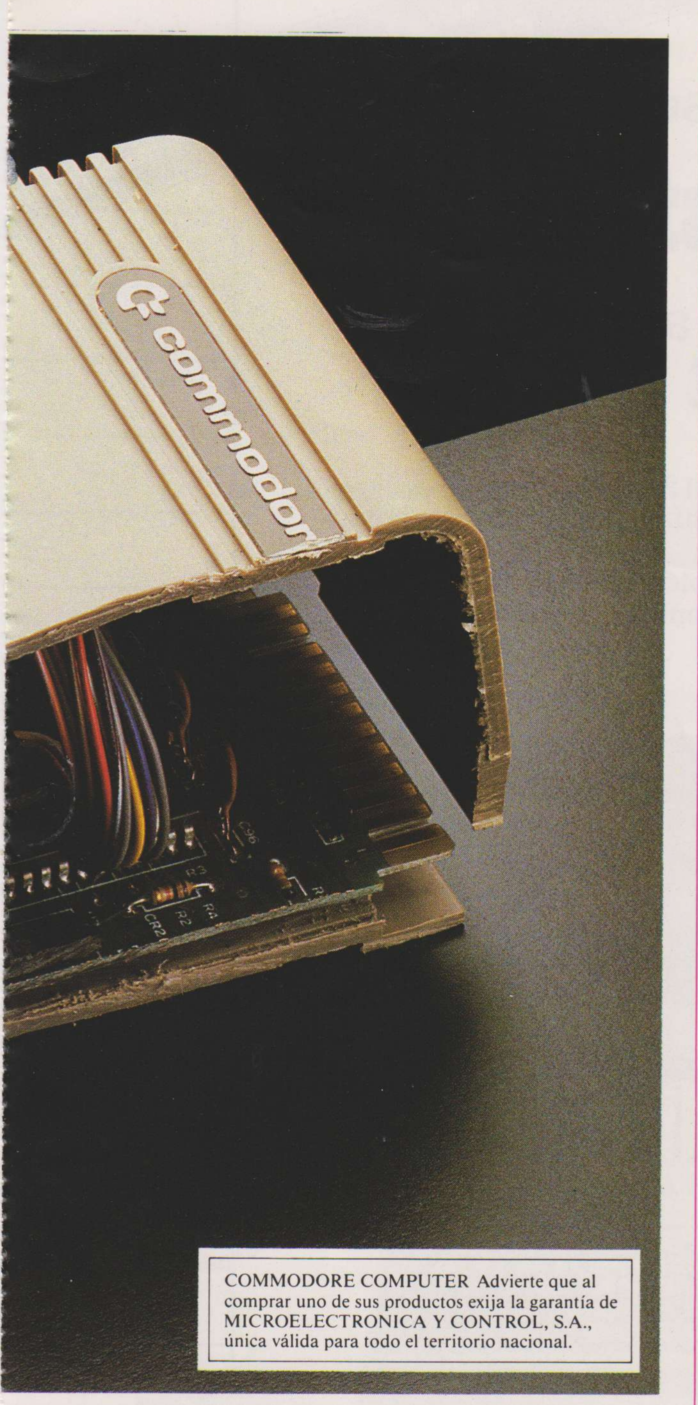
Utilizar un COMMODORE 64 únicamente para jugar, es como pedirle a Albert Einstein que nos resuelva la raíz cuadrada de cuatro.

El cerebro del ordenador apenas funciona.

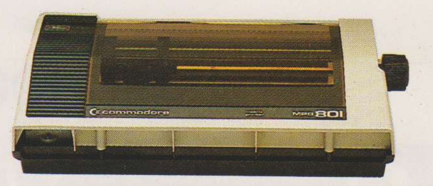
Para aprovecharlo al máximo, lo que Vd. necesita son programas interesantes (archivo de documentos, educación interactiva, estimulantes juegos o proceso de textos) y lógicamente los periféricos adecuados:

— La unidad de disco Commodore, un almacenaje de datos muy rápido, fiable y de gran capacidad.
— La unidad de cassette Commodore, una forma económica de introducir y almacenar programas.
— El monitor en color Commodore, para

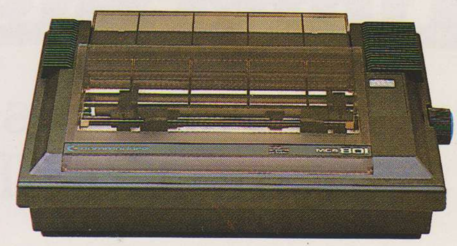
MICROELECTRONICA Y CONTROL c/ Valencia, 49-53. 08015 Barcelona. c/ Princesa, 47, 3.º G. 28008 Madrid.
UNICO REPRESENTANTE DE COMMODORE EN ESPAÑA



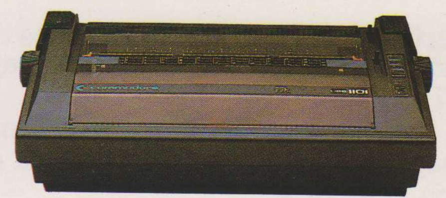
COMMODORE COMPUTER Advierte que al comprar uno de sus productos exija la garantía de MICROELECTRONICA Y CONTROL, S.A., única válida para todo el territorio nacional.



COMMODORE MPS 801
Impresora de matriz



COMMODORE MCS 801
Impresora cuatro colores



COMMODORE DPS 1101
Impresora de margarita

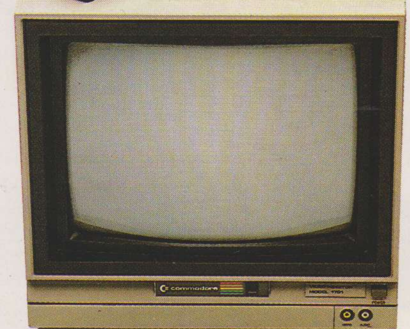


COMMODORE 1520
Printer plotter cuatro colores

COMMODORE 1541
Unidad de disco 170 K



COMMODORE 1530
Unidad de cassette



COMMODORE 1701
Monitor con color y sonido

JOYSTICKS
PADDLES
SOFTWARE Y LIBROS

10% de su cerebro?

lograr que los textos y gráficos sean más claros y vivos que en un televisor.

— 3 impresoras y un printer-plotter, para copias en papel, en blanco y negro o color.

— Los joystick y paddles, para conseguir que los juegos sean más excitantes.

Sea exigente. Si su cerebro, el de su COMMODORE 64, puede darle 100, no se conforme con 10.



la emoción del futuro

Todos los meses publicaremos programas para los Commodores.

Suscríbase a

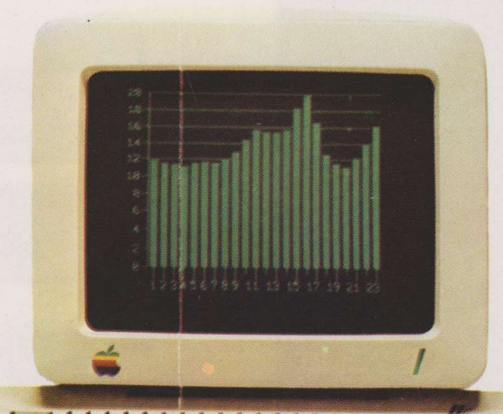
MicroSistemas

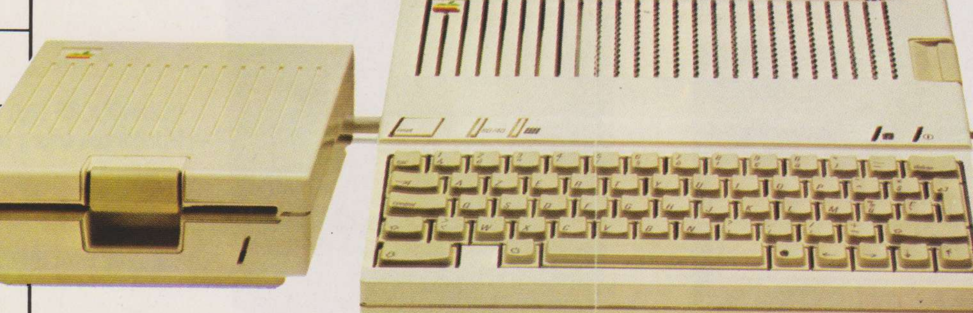
y le llevaremos a su casa un APPLE II c

Suscríbase a MicroSistemas antes del 30 de noviembre próximo y participará en el sorteo de un magnífico ordenador personal APPLE II c.

Para los que ya estén suscritos también tendrán opción de conseguir el APPLE II c. Si desea conocer su número en el sorteo llámenos al teléfono 231 23 95.

Habrá también otros premios para nuestros suscriptores que daremos oportunamente a conocer. El sorteo tendrá lugar en nuestro stand del SIMO.





BOLETIN DE SUSCRIPCION A MICROSISTEMAS

Nombre y Apellidos ....

Puesto ....

Compañía ....

Actividad Empresa ....

Dirección .... Teléf.: ....

Ciudad .... D. P. .... Provincia ....

Suscripción anual 2.950 ptas.

Deseo iniciar la suscripción con el próximo número

☐ Adjunto cheque por 2.950 ptas.

☐ Reembolso más gastos del mismo al recibir el primer número de la suscripción.

(Enviar a Gravina, 13-1º Izda. Madrid-4 - Teléf.: 231 23 88/95)

```
182 CT=CT+1:POKES+24,0:POKE53269,0
183 PRINT"[21CRSRD]";
184 IFCT/2=INT(CT/2)THENPRINT"[RED]J
UGADOR[SPC]1[SPC]=";J1;"[9CRSRR][YEL]
JUGADOR[SPC]2[SPC]=";J2;"[RED][HOM]"
;
185 IFCT/2<>INT(CT/2)THENPRINT"[YEL]
JUGADOR[SPC]1[SPC]=";J1;"[9CRSRR][RED]
JUGADOR[SPC]2[SPC]=";J2;"[RED][HOM]"
;
186 PRINT"[3SPC]A[3SPC]B[3SPC]C[3SPC]
D[3SPC]E[3SPC]F[3SPC]G[3SPC]H[3SPC]I
"
190 FOR N=1TO15
200 PRINT"[COMM8][2SPC][RVSON][3SPC]
[RVSOFF][SPC][RVSON][3SPC][RVSOFF][SPC]
[RVSON][3SPC][RVSOFF][SPC][RVSON][3SPC]
[RVSOFF][SPC][RVSON][3SPC][RVSOFF][SPC]
[RVSON][3SPC]";
202 PRINT"[RVSOFF][SPC][RVSON][3SPC]
[RVSOFF][SPC][RVSON][3SPC][RVSOFF][SPC]
[RVSON][3SPC][RVSOFF]"
210 IFN/3=INT(N/3)THENPRINT
250 NEXT
260 PRINT"[HOM][RED][2CRSRD]1[CRSRL]
[4CRSRD]2[CRSRL][4CRSRD]3[CRSRL][4CRSRD]
4[CRSRL][4CRSRD]5[4CRSRD][YEL]"
263 GOSUB2500:POKE53269,0
270 IFH=0THEN1000
275 IFCT/2=INT(CT/2)THENPRINTTAB(22)
"[SPC]";
276 PRINT"OBJETO[SPC]1:[SPC]";
277 GETL1$:IFL1$=""THEN277
279 IFASC(L1$)<65ORASC(L1$)>73THEN27
7
280 PRINTL1$;"-";
282 GETN1$:IFN1$=""THEN282
283 IFASC(N1$)<49ORASC(N1$)>53THEN28
2
284 PRINTN1$:POKEV+28,3
320 L1=ASC(L1$)-64:N1=VAL(N1$):FORE=
0TO30:NEXT
322 PRINT"[CRSRU][37SPC]":PRINT"[CRSRU]
";
325 IFO(L1,N1)=0THEN275
330 POKE2040,229+O(L1,N1):O1=O(L1,N1
)
340 C=39:P1=0:CX=0:CY=1:O=1:GOSUB110
0:POKEV+21,1
360 IFCT/2=INT(CT/2)THENPRINTTAB(22)
"[SPC]";
361 PRINT"OBJETO[SPC]2:[SPC]";
362 GETL2$:IFL2$=""THEN362
363 IFASC(L2$)<65ORASC(L2$)>73THEN36
2
365 PRINTL2$;"-";
366 GETN2$:IFN2$=""THEN366
368 IFASC(N2$)<49ORASC(N2$)>53THEN36
6
370 PRINTN2$:FORE=0TO30:NEXT
372 PRINT"[CRSRU][36SPC]":PRINT"[CRSRU]
";
400 POKEV+21,0:L2=ASC(L2$)-64:N2=VAL
(N2$)
403 IF(L2=L1)AND(N2=N1)THEN360
405 IFO(L2,N2)=0THEN361
410 POKE2041,229+O(L2,N2)
415 O2=O(L2,N2)
420 C=40:CX=2:CY=3:L1=L2:N1=N2:O=2:G
OSUB1100
430 POKEV+21,2:IFO1=O2THEN2000
435 PRINT"LO[SPC]SIENTO[SPC]TIO[SPC]
FALLASTE......"
440 FORL=STOS+24:POKEL,0:NEXT
450 POKES+5,96:POKES+6,200:POKES+24,
79
470 FORA=0TO50STEP3:HF=PEEK(14592+A)
:LF=PEEK(14593+A)
480 DR=PEEK(14594+A)*4
490 POKES+1,HF:POKES,LF:POKES+4,33
```

```
510 FORT=1TODR:NEXT:POKES+4,32:FORT=
1TO50:NEXT:NEXT
530 PRINT"[CRSRU][28SPC]":PRINT"[CRSRU]
";
540 GOTO180
999 END
1000 DIMO(9,5)
1005 PRINT"UN[SPC]MINUTO[SPC]PARA[SPC]
COLOCAR[SPC]LOS[SPC]DIBUJOS[5SPC][CRSRU]
"
1010 FOR N=1 TO9:FOR M=1TO5
1020 AL=INT(RND(0)*45)+1
1030 FORP=1TON:FORQ=1TO5
1033 IFAL=O(P,Q)THEN1020
1035 NEXTQ:NEXTP:O(N,M)=AL:NEXTM:NEX
TN
1052 FORN=1TO9:FORM=1TO5
1053 IFO(N,M)>23THENO(N,M)=O(N,M)-23
1054 NEXT:NEXT:H=1
1058 PRINT"[39SPC][CRSRU]"
1060 GOTO275
1100 X1=40+(32*(L1-1)):Y1=60+(32*(N1
-1))
1110 IF(O=1)AND(X1>255)THENP1=1
1115 IF(O=2)AND(X1>255)THENP1=P1+2
1130 IFX1>255THENX1=X1-256
1140 POKEV+CX,X1:POKEV+16,P1:POKEV+C
Y,Y1
1150 POKEV+37,PEEK(16193+3*(O(L1,N1)
-1))
1160 POKEV+38,PEEK(16194+3*(O(L1,N1)
-1))
1170 POKEV+C,PEEK(16195+3*(O(L1,N1)-
1))
1180 FORL=STOS+24:POKEL,0:NEXT
1190 POKES+5,9:POKES+6,32
1200 POKES+24,15:POKES+1,60:POKES,20
0
1210 POKES+4,33:FORN=0TO50:NEXT:POKE
S+24,0
1220 RETURN
2000 IFCT/2=INT(CT/2)THENJ2=J2+1
2010 IFCT/2<>INT(CT/2)THENJ1=J1+1
2020 PRINT"MUY[SPC]BIEN[SPC]CONSEGUI
STE[SPC]UNA[SPC]PAREJA"
2030 POKE53269,3:S=54272
2040 FORL=STOS+24:POKEL,0:NEXT
2050 POKES+3,15:POKES+2,15:POKES+5,8
8:POKES+6,89:POKES+24,15
2060 FORQ=0TO36STEP3:HF=PEEK(14644+Q
):LF=PEEK(14645+Q):DR=PEEK(14646+Q)*
4
2065 POKE53269,3
2070 POKES+1,HF:POKES,LF:POKES+4,65:
FORT=1TODR:NEXT:POKE53269,0
2080 POKES+4,64:FORT=1TO50:NEXT
2085 NEXTQ
2140 IFJ1+J2=0THEN2590
2150 NU=J1+J2
2155 IFNU=22THEN3000
2160 X1(NU)=ASC(L1$)-64:Y1(NU)=VAL(N
1$)
2170 X2(NU)=L1:Y2(NU)=N1
2380 O(ASC(L1$)-64,VAL(N1$))=0:O(L1,
N1)=0
2390 PRINT"[CRSRU][33SPC]":PRINT"[CRSRU]
"
2400 CT=CT-1:GOTO180
2500 IFNU=0THENRETURN
2510 FORWS=1TONU
2512 PRINT"[HOM]";
2515 FORN=1TO((Y1(WS)-1)*4)+1:PRINT"
[CRSRD]";:NEXT
2520 PRINTTAB(((X1(WS)-1)*4)+2)"[COMM1]
[RVSON][3SPC][RVSOFF]";:PRINTSPC(37)
"[RVSON][3SPC][RVSOFF]";
2525 PRINTSPC(37)"[RVSON][3SPC][RVSOFF]
";
```

```
2530 PRINT"[HOM]";
2540 FORN=1TO((Y2(WS)-1)*4)+1:PRINT"
[CRSRD]";:NEXT
2550 PRINTTAB(((X2(WS)-1)*4)+2)"[COMM1]
[RVSON][3SPC][RVSOFF]";:PRINTSPC(37)
"[RVSON][3SPC][RVSOFF]";
2555 PRINTSPC(37)"[RVSON][3SPC][RVSOFF]
";
2557 PRINT"[HOM]";
2570 NEXT
2590 PRINT"[23CRSRD]";"[YEL]";
2900 RETURN
3000 IFJ1>J2THENA$="GANA[SPC]EL[SPC]
JUGADOR[SPC]N.[SPC]1"
3010 IFJ2>J1THENA$="GANA[SPC]EL[SPC]
JUGADOR[SPC]N.[SPC]2"
3020 POKE53280,5:POKE53281,5:PRINT"[CLR]
[COMM2]";
3030 IFJ1=J2THEN3060
3040 PRINT"[8CRSRD][11CRSRR]ENHORABU
ENA"
3050 PRINT"[4CRSRD][7CRSRR]";A$
3055 IFJ1<>J2THEN3070
3060 PRINT"[11CRSRD][6CRSRR]POR[SPC]
ESTA[SPC]VEZ[SPC]EMPATASTEIS"
3070 PRINT"[5CRSRD][WHT]QUEREIS[SPC]
PROBAR[SPC]OTRA[SPC]VEZ[SPC]S/N?"
3080 GETB$:IFB$="S"THENRUN18
3085 IFB$=""THEN3080
3090 PRINT"[CLR]":END
14000 FORN=14720TO16191:READA:POKEN,
A:NEXT
14010 FORN=16193TO16262:READA:POKEN,
A:NEXT
14020 FORN=14592TO14681:READA:POKEN,
A:NEXT
14030 RETURN
15000 REM***********BARBERO**********
**
15010 DATA0,0,0,17,0,1,85,80,5,89,
84,4,170
15020 DATA132,22,170,165,18,170,161,
14,106,108,14,174,172,14
15030 DATA191,172,14,191,172,2,170,1
60,2,153,160,2,170,160
15040 DATA0,187,128,0,174,128,0,42,0
,0,42,0,0,8
15050 DATA0,0,0,0,0,0,0,0
15060 REM**********PINO**************
***
15070 DATA0,0,0,4,0,0,21,0,0,85,64
,0,21
15080 DATA0,0,85,64,1,85,80,0,85,64,
1,85,80,5
15090 DATA85,84,1,85,80,5,85,84,21,8
5,85,0,8,0
15100 DATA0,8,0,0,8,0,0,8,0,0,8,0,0,
8
15110 DATA0,0,0,0,0,0,0,0
15120 REM**********PAYASO**********
***
15130 DATA0,21,0,0,21,0,0,21,0,1,85,
80,15,170
15140 DATA188,62,170,175,62,187,175,
62,235,239,14,171,172,14
15150 DATA166,172,14,149,172,14,166,
172,2,170,160,2,170,160
15160 DATA0,187,128,0,174,128,0,42,0
,0,42,0,0,8
15170 DATA0,0,0,0,0,0,0,0
15180 REM**********MAZINGER*********
***
15190 DATA21,85,84,23,215,212,21,85,
84,21,85,84,90,170
15200 DATA165,90,170,165,95,255,245,
95,255,245,223,255,247,223
15210 DATA255,247,23,235,164,26,235,
164,26,190,164,26,190,164
15220 DATA58,170,172,10,170,160,14,1
90,176,4,190,128,2,170
15230 DATA192,0,170,0,0,203,0,0
15240 REM**********CASITA***********
***
15250 DATA5,87,64,6,87,80,6,87,80,26
,149,84,26,149
15260 DATA84,107,165,85,107,165,85,1
70,170,170,170,170,170,170
15270 DATA171,234,170,171,234,175,17
1,234,175,171,234,175,170,170
15280 DATA175,170,170,175,170,170,17
5,170,170,15,0,0,3,192
15290 DATA0,0,240,0,0,60,0,255
15300 REM**********BARCO************
***
15310 DATA 0, 0, 0, 0, 0, 48, 0, 0,
48, 48, 0, 240, 48, 0
15320 DATA 192, 16, 3, 0, 48, 21, 0,
 48, 21, 0, 16, 63, 0, 240
15330 DATA 63, 0, 252, 21, 0, 207, 6
3, 0, 255, 63, 0, 85, 85, 84
15340 DATA 85, 85, 80, 85, 85, 80, 2
1, 85, 80, 5, 85, 96, 170, 170
15350 DATA 170, 170, 170, 170, 170,
170, 170, 255
15360 REM***********PATO*************
***
15370 DATA 0, 0, 0, 0, 80, 0, 1, 84,
 0, 41, 212, 0, 9, 80
15380 DATA 0, 0, 64, 0, 0, 64, 0, 0,
 64, 0, 1, 64, 0, 5
15390 DATA 64, 0, 5, 80, 0, 21, 84,
0, 21, 85, 80, 21, 85, 64
15400 DATA 21, 85, 0, 5, 84, 0, 1, 8
0, 0, 8, 8, 0, 40, 10
15410 DATA 0, 255, 255, 255, 255, 25
5, 255, 255
15420 REM**********MANZANA**********
***
15430 DATA 0, 2, 0, 1, 8, 16, 1, 8,
16, 5, 72, 84, 5, 72
15440 DATA 84, 21, 85, 85, 21, 85, 8
5, 21, 85, 85, 21, 85, 85, 5
15450 DATA 85, 84, 5, 86, 148, 5, 86
, 148, 1, 87, 176, 1, 86, 220
15460 DATA 1, 85, 83, 1, 85, 83, 0,
85, 64, 0, 81, 64, 0, 81
15470 DATA 64, 0, 17, 0, 0, 0, 0, 0
15480 REM**********LAMPARA**********
***
15490 DATA 0, 16, 0, 0, 84, 0, 0, 22
0, 0, 1, 117, 0, 1, 85
15500 DATA 0, 5, 85, 64, 5, 85, 64,
21, 85, 80, 55, 119, 96, 93
15510 DATA 221, 212, 85, 85, 84, 0,
168, 192, 0, 168, 192, 2, 170, 192
15520 DATA 10, 170, 192, 42, 170, 22
4, 42, 170, 160, 10, 170, 128, 2, 17
0
15530 DATA 0, 2, 170, 0, 2, 170, 0,
255
15540 REM***********CHINO************
***
15550 DATA 0, 0, 0, 0, 4, 0, 0, 21,
0, 0, 85, 64, 0, 85
15560 DATA 64, 1, 85, 80, 5, 85, 84,
5, 85, 84, 21, 85, 85, 58
15570 DATA 170, 168, 58, 234, 232, 1
4, 187, 160, 50, 170, 160, 14, 170,
160
15580 DATA 50, 170, 160, 12, 187, 12
8, 48, 174, 128, 12, 42, 0, 48, 42
15590 DATA 0, 48, 8, 0, 0, 0, 0, 255

15600 REM**********FLORERO**********
***
15610 DATA 0, 0, 0, 0, 0, 0, 1, 1, 0
, 5, 69, 68, 7, 3
15620 DATA 65, 3, 51, 13, 51, 12, 51
```

```
, 207, 204, 192, 0, 51, 0, 63
15630 DATA 51, 0, 51, 51, 0, 218, 17
0, 160, 242, 170, 128, 80, 170, 0
15640 DATA 0, 170, 0, 0, 170, 0, 0,
170, 0, 0, 170, 0, 32, 40
15650 DATA 8, 42, 170, 168, 42, 170,
 168, 255
15660 REM**********COCHE************
***
15670 DATA 0, 0, 0, 0, 0, 0, 1, 85,
64, 1, 85, 64, 1, 16
15680 DATA 64, 1, 16, 64, 1, 16, 64,
 1, 16, 64, 5, 16, 64, 5
15690 DATA 16, 64, 5, 85, 84, 5, 85,
 84, 7, 85, 212, 215, 85, 215
15700 DATA 223, 215, 247, 13, 195, 1
12, 3, 0, 192, 3, 0, 192, 170, 170
15710 DATA 170, 170, 170, 170, 170,
170, 170, 255
15720 REM***********CARACOL*********
***
15730 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0
, 0, 0, 0, 0, 192
15740 DATA 0, 3, 64, 0, 61, 112, 0,
213, 80, 0, 213, 92, 0, 221
15750 DATA 92, 51, 215, 92, 34, 215,
 92, 34, 213, 215, 34, 213, 215, 10
15760 DATA 23, 87, 2, 62, 87, 40, 42
, 254, 168, 170, 170, 160, 85, 169
15770 DATA 168, 85, 85, 85, 85, 85,
85, 255
15780 REM**********TETERA***********
***
15790 DATA 0, 0, 0, 0, 12, 0, 0, 4,
0, 0, 21, 1, 0, 85
15800 DATA 65, 5, 255, 197, 5, 85, 6
9, 20, 85, 68, 20, 85, 68, 81
15810 DATA 85, 100, 81, 85, 100, 65,
 85, 100, 65, 85, 100, 69, 85, 92
15820 DATA 69, 85, 92, 85, 85, 84, 8
5, 85, 84, 21, 85, 85, 21, 85
15830 DATA 85, 0, 0, 0, 0, 0, 0, 255

15840 REM**********MANZANO**********
***
15850 DATA 0, 20, 0, 1, 85, 64, 5, 8
5, 80, 7, 87, 80, 23, 87
15860 DATA 84, 21, 85, 84, 21, 117,
92, 21, 117, 92, 29, 85, 212, 29
15870 DATA 85, 212, 5, 117, 80, 5, 1
17, 80, 5, 105, 80, 1, 105, 112
15880 DATA 0, 40, 48, 0, 40, 0, 0, 4
0, 0, 0, 40, 0, 0, 40
15890 DATA 0, 85, 105, 85, 85, 105,
85, 255
15900 REM***********CARA PALO*******
***
15910 DATA 3, 255, 240, 3, 255, 240,
 3, 255, 240, 63, 255, 255, 1, 85
15920 DATA 80, 3, 255, 240, 1, 247,
208, 1, 247, 208, 1, 85, 80, 1
15930 DATA 85, 80, 1, 93, 80, 1, 127
, 80, 1, 85, 80, 1, 106, 80
15940 DATA 1, 85, 80, 0, 21, 0, 0, 2
9, 0, 0, 63, 0, 0, 63
15950 DATA 0, 0, 0, 0, 0, 0, 0, 255
15960 REM***********MOLINO**********
***
15970 DATA 0, 0, 0, 0, 20, 0, 0, 85,
 0, 48, 85, 12, 13, 85
15980 DATA 112, 3, 85, 192, 1, 215,
64, 1, 125, 64, 2, 190, 128, 2
15990 DATA 235, 128, 3, 170, 192, 14
, 170, 176, 50, 170, 140, 2, 170, 12
8
16000 DATA 2, 170, 128, 2, 170, 128,
 2, 150, 128, 2, 150, 128, 2, 150
16010 DATA 128, 2, 150, 128, 2, 150,
 128, 255
```

```
16020 REM***********NINA************
***
16030 DATA 0, 0, 0, 0, 0, 0, 5, 69,
64, 5, 85, 64, 21, 85
16040 DATA 80, 21, 101, 80, 22, 170,
 80, 22, 170, 80, 26, 170, 144, 26
16050 DATA 170, 144, 25, 101, 144, 2
6, 170, 144, 26, 170, 144, 26, 170,
144
16060 DATA 90, 170, 148, 90, 186, 14
8, 10, 186, 128, 2, 170, 0, 0, 168
16070 DATA 0, 0, 32, 0, 0, 0, 0, 255

16080 REM**********CIGARRO**********
***
16090 DATA 0, 0, 0, 0, 0, 10, 0, 0,
10, 0, 0, 170, 0, 0
16100 DATA 42, 0, 0, 42, 0, 0, 128,
0, 0, 128, 0, 2, 0, 0
16110 DATA 0, 0, 0, 40, 0, 0, 168, 0
, 2, 128, 0, 2, 128, 0
16120 DATA 8, 0, 0, 32, 0, 0, 62, 17
0, 149, 62, 170, 149, 62, 170
16130 DATA 149, 0, 0, 0, 0, 0, 0, 25
5
16140 REM**********PAJARITA*********
***
16150 DATA 0, 29, 221, 0, 55, 116, 0
, 29, 208, 0, 55, 112, 0, 29
16160 DATA 192, 0, 55, 64, 0, 29, 19
2, 0, 55, 112, 0, 29, 220, 0
16170 DATA 55, 116, 0, 29, 221, 0, 1
19, 119, 0, 221, 221, 3, 119, 119
16180 DATA 13, 221, 221, 7, 116, 55,
 29, 220, 13, 119, 112, 3, 221, 208
16190 DATA 1, 0, 0, 0, 0, 0, 0, 255
16200 REM**********PARAGUAS*********
***
16210 DATA 0, 40, 0, 0, 40, 0, 2, 17
0, 128, 10, 170, 160, 42, 170
16220 DATA 168, 170, 170, 170, 170,
174, 170, 128, 140, 130, 128, 140, 1
30, 128
16230 DATA 140, 130, 0, 12, 0, 0, 12
, 0, 0, 12, 0, 0, 12, 0
16240 DATA 0, 12, 0, 0, 12, 0, 0, 12
, 0, 0, 68, 0, 1, 4
16250 DATA 0, 1, 4, 0, 0, 84, 0, 255

16260 REM**********CABALLITO********
***
16270 DATA 0, 0, 0, 0, 0, 0, 0, 128,
 0, 2, 0, 0, 170, 0
16280 DATA 0, 42, 0, 0, 170, 0, 0, 4
2, 0, 0, 10, 191, 232, 10
16290 DATA 191, 226, 10, 191, 226, 1
0, 170, 162, 2, 170, 162, 0, 136, 12
8
16300 DATA 0, 128, 128, 0, 128, 32,
2, 0, 8, 72, 0, 9, 88, 0
16310 DATA 5, 21, 85, 84, 5, 85, 80,
 255
16320 REM***********AVION***********
***
16330 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0
, 0, 0, 10, 0, 0
16340 DATA 40, 0, 0, 160, 0, 2, 130,
 192, 74, 2, 193, 90, 2, 202
16350 DATA 170, 170, 42, 170, 170, 2
02, 170, 168, 194, 170, 160, 194, 12
8, 0
16360 DATA 10, 0, 0, 40, 0, 0, 160,
0, 0, 0, 0, 0, 0, 0
16370 DATA 0, 0, 0, 0, 0, 0, 0, 255
16400 REM*****COLORES SPRITES*****
****
16420 DATA0,2,10,5,15,9,9,0,10,2,0,8
,2,0
16430 DATA1,2,0,6,8,0,2,10,0,2,5,0,6
```

```
,2
16440 DATA0,7,2,5,9,14,0,9,5,0,8,4,0
,0
16450 DATA5,2,9,8,0,2,2,0,1,9,2,8,9,
11
16460 DATA12,6,2,0,8,10,14,11,10,2,1
,0,14,255
16500 REM******MUSICA**************
***
16530 DATA15,197,62,18,234,62,15,197
,62,15,197,62,15,197,31,18,234,31,22
,114
16540 DATA187,15,197,62,18,234,62,9,
83,62,9,83,62,9,83,62,11,174,15,14,8
3
16550 DATA15,15,197,62,14,83,31,9,83
,62,255,34,75,62,43,52,62,51,97,93,4
3,52
16560 DATA31,51,97,62,57,172,62,51,9
7,125,0,0,31,43,52,62,51,97,62,57,17
2
16570 DATA250,51,97,125,0,0
```

Triángulos

Ricardo Montañana Gómez
Horno, nº 4-1º
Madrigueras (Albacete)

VIC-20

TRIANGULOS proporciona una solución de ángulos, lados y funciones trigonométricas de los ángulos por pantalla y por impresora (opcional).

El programa se basa en un triángulo de forma:

pidiendo siempre tres datos y devolviéndonos los otros tres y sus funciones trigonométricas.

El programa está estructurado de la siguiente forma:

Líneas 1-80 Presentación del programa (menú principal de opciones)
Líneas 90-100 Ejecuta la subrutina indicada por el menú.
Líneas 1000-1110 Subrutina de 2 lados y ángulo comprendido.
Líneas 2000-2110 Subrutina de 1 lado y ángulos adyacentes.
Líneas 3000-3110 Subrutina de 2 lados y ángulo no comprendido.
Líneas 4000-4100 Subrutina de un lado 1 ángulo adyacente y el ángulo opuesto.
Líneas 6000-6043 Subrutina de comprobación de los tres lados (para saber si forman un triángulo las cantidades introducidas o no).
Líneas 6100-6190 Cálculo de los resultados con los tres lados.
Líneas 7000-7070 Subrutina de salida. Aquí se pide la salida o no de los datos por impresora y si se quiere acabar ya con el programa.
Líneas 9000-9060 Subrutina de salida de datos por pantalla.
Líneas 10000-11150 Subrutina de salida de datos por impresora.

El programa ocupa 4434 bytes de memoria, por lo tanto no se puede ejecutar sin disponer de ampliación de memoria. Aunque sí se podría ejecutar con la memoria estandard si se quita la subrutina de salida de datos por impresora, el programa funcionará igual de bien, ya que la salida de datos por este periférico es opcional.

La adaptación de este programa a un 64 creo que es muy sencilla, tan sólo se trata de cambiar el POKE de la línea 10 (este POKE cambia el color de la pantalla).

```
10 POKE36879,123:PRINTCHR$(14):PRINT
 "[CLR]"
20 PRINT "[HOM][RVSON][PUR][6SPC][SHIFTT]
[SHIFTR][SHIFTI][SHIFTA][SHIFTN][SHIFTG]
[SHIFTU][SHIFTL][SHIFTO][SHIFTS][6SPC]
[RVSOFF]"
30 PRINT "[2CRSRD]1...[SPC][RVSON][WHT]
2[PUR][SPC]LADOS[SPC]Y[SPC]ANGULO":P
RINT"[5CRSRR][RVSON][2SPC]COMPRENDID
O."
40 PRINT "[CRSRD]2...[SPC][WHT][RVSON]
1[PUR][SPC]LADO[2SPC]Y[SPC][WHT]2[PUR]
[SPC]ANGU-[5CRSRR][2SPC]LOS[SPC]ADYA
CENTES.[2CRSRU]"
50 PRINT "3...[SPC][WHT][RVSON]2[PUR]
[SPC]LADOS[SPC]Y[SPC]ANGULO":PRINT"[5CRSRR]
[RVSON][2SPC]NO[SPC]COMPRENDIDO"
60 PRINT"[CRSRD]4...[SPC][WHT][RVSON]
1[PUR][SPC]LADO[SPC]UN[SPC]ANGULO"
61 PRINT "[5CRSRR][RVSON][2SPC]ADYAC
ENTE[SPC]Y[SPC]EL[RVSOFF][6SPC][RVSON]
[2SPC]ANGULO[SPC]OPUESTO."
63 , FILE EXISTS,00,00
70 PRINT "5...[SPC][RVSON]TRES[SPC]L
ADOS.
80 PRINT "[2CRSRD]6...[SPC][RVSON]SA
LIDA.[BLU]"
90 GETA$:S=VAL(A$):IFS>7ORS<1THEN 90

100 ON S GOSUB 1000,2000,3000,4000,6
000,7000:GOTO 10
1000 REM **** 2 LADOS Y ANGULO COMPR
ENDIDO ****
1010 PRINT "[CLR][WHT][RVSON][SPC][RED]
2[WHT][SPC][SHIFTL][SHIFTA][SHIFTD][SHIFTO]
[SHIFTS][SPC][SPC][SHIFTY][SPC][SHIFTA]
[SHIFTN][SHIFTG][SHIFTU][SHIFTL][SHIFTO]
[8SPC][SHIFTC][SHIFTO][SHIFTM][SHIFTP]
[SHIFTR][SHIFTE][SHIFTN][SHIFTD][SHIFTI]
[SHIFTD][SHIFTO].[6SPC][PUR]"
1020 INPUT "ANGULO[SPC]´[SHIFTA]´";A
1
1030 INPUT "LADOS[SPC]´B´[SPC]Y[SPC]
´C´[SPC]";B,C
1040 AA=π*A1/180
1050 A=ABS(SQR(B↑2+C↑2-(2*B*C*COS(AA
))))
1060 SENC=SIN(AA)*C/A:CC=ATN(SENC/SQ
R(-SENC*SENC+1)):C1=CC*180/π
1070 B1=180-A1-C1:BB=B1*π/180
1080 GOSUB 9000
1090 IF SW=1 THEN GOSUB 10000
1100 GET A$:IF A$<>"R" THEN 1100
1110 RETURN
2000 REM **** 1 LADO Y ANGULOS ADYAC
ENTES ****
2010 PRINT "[CLR][WHT][RVSON][SPC][RED]
1[WHT][SPC][SHIFTL][SHIFTA][SHIFTD][SHIFTO]
```

```
[SPC][SHIFTY][SPC][RED]2[WHT][SPC][SHIFTA]
[SHIFTN][SHIFTG][SHIFTU][SHIFTL][SHIFTO]
[SHIFTS][SPC][7SPC][SHIFTA][SHIFTD][SHIFTY]
[SHIFTA][SHIFTC][SHIFTE][SHIFTN][SHIFTT]
[SHIFTE][SHIFTS].[6SPC][PUR]"
2020 INPUT "ANGULOS[SPC]´[SHIFTC]´[SPC]
Y[SPC]´[SHIFTB]´[SPC]";C1,B1
2040 INPUT "LADO[SPC]´A´[SPC]";A
2050 A1=180-B1-C1:AA=π*A1/180:BB=π*B
1/180:CC=π*C1/180
2060 B=A*SIN(BB)/SIN(AA)
2070 C=A*SIN(CC)/SIN(AA)
2080 GOSUB 9000
2090 IF SW=1 THEN GOSUB 10000
2100 GET A$:IFA$<>"R"THEN 2100
2110 RETURN
3000 REM **** 2 LADOS Y ANGULO NO CO
MPRENDIDO ****
3010 PRINT "[CLR][WHT][RVSON][SPC][RED]
2[WHT][SPC][SHIFTL][SHIFTA][SHIFTD][SHIFTO]
[SHIFTS][SPC][SHIFTY][SPC][SHIFTA][SHIFTN]
[SHIFTG][SHIFTU][SHIFTL][SHIFTO][SPC]
[SHIFTN][SHIFTO][6SPC][SHIFTC][SHIFTO]
[SHIFTM][SHIFTP][SHIFTR][SHIFTE][SHIFTN]
[SHIFTD][SHIFTI][SHIFTD][SHIFTO].[6SPC]
[PUR]"
3020 INPUT "ANGULO[SPC]´[SHIFTB]´[SPC]
";B1
3030 INPUT "LADOS[SPC]´B´[SPC]Y[SPC]
´C´[SPC]";B,C
3040 BB=π*B1/180
3050 SENC=C*SIN(BB)/B:CC=ATN(SENC/SQ
R(-SENC*SENC+1)):C1=180*CC/π
3060 A1=180-B1-C1:AA=A1*π/180
3070 A=B*SIN(AA)/SIN(BB)
3080 GOSUB 9000
3090 IF SW=1 THEN GOSUB 10000
3100 GET A$:IF A$<>"R"THEN 3100
3110 RETURN
4000 REM **** 1 LADO EL ANGULO OPUES
TO Y EL ANGULO ADYACENTE ****
4010 PRINT "[CLR][WHT][RVSON][SPC][RED]
1[WHT][SPC][SHIFTL][SHIFTA][SHIFTD][SHIFTO]
[SPC][RED]1[WHT][SPC][SHIFTA][SHIFTN]
[SHIFTG][SHIFTU][SHIFTL][SHIFTO][SPC]
[SHIFTA][SHIFTD][SHIFTY][SHIFTA]-[SHIFTC]
[SHIFTE][SHIFTN][SHIFTT][SHIFTE][SPC]
[SHIFTY][SPC][SHIFTA][SHIFTN][SHIFTG]
[SHIFTU][SHIFTL][SHIFTO][SPC][SHIFTO]
[SHIFTP][SHIFTU][SHIFTE][SHIFTS][SHIFTT]
[SHIFTO][PUR]"
4020 INPUT "ANGULOS[SPC]´[SHIFTC]´[SPC]
Y[SPC]´[SHIFTA]´[SPC]";C1,A1
4030 INPUT "LADO[SPC]´A´[SPC]";A
4040 B1=180-C1-A1:AA=A1*π/180:BB=B1*
π/180:CC=C1*π/180
4050 C=A*SIN(CC)/SIN(AA)
4060 B=A*SIN(BB)/SIN(AA)
4070 GOSUB 9000
4080 IF SW=1 THEN GOSUB 10000
4090 GET A$:IF A$<>"R"THEN 4090
4100 RETURN
6000 REM **** TRES LADOS. ****
6010 PRINT "[CLR][WHT][RVSON][5SPC][SHIFTC]
[SHIFTO][SHIFTM][SHIFTP][SHIFTR][SHIFTO]
[SHIFTB][SHIFTA][SHIFTC][SHIFTI][SHIFTO]
[SHIFTN][5SPC][RVSOFF][PUR]"
6020 INPUT "[CRSRD]LADO[SPC]1[COMMV]
:";A:INPUT "[CRSRD]LADO[SPC]2[COMMV]
:";B:INPUT "[CRSRD]LADO[SPC]3[COMMV]
:";C
6030 P=(A+B+C)/2:S=(P*(P-A)*(P-B)*(P
-C))
6031 IF S>0.0THEN PRINT "[2CRSRD][RVSON]
[WHT][SPC]TODO[SPC][SHIFTO].[SHIFTK]
[SPC][PUR]":PRINT "[2CRSRD][RVSON][WHT]
[SPC]PULSA[SPC]´[SHIFTR]´[SPC][PUR]"
:GOTO 6042
6040 PRINT "[BLK]ESE[SPC]TRIANGULO[SPC]
NO[SPC]EXIS-TE":PRINT "MALA[SPC]SUER
TE.":PRINT "[SPC]PULSA[SPC]LA[SPC]´[SHIFTR]
´":SH=1
6042 GET A$:IFA$<>"R"THEN6042
6043 IFSH=1THENSH=0:GOTO 10
6100 PRINT "[CLR][WHT][RVSON][5SPC][SHIFTT]
[SHIFTR][SHIFTE][SHIFTS][SPC][SHIFTL]
[SHIFTA][SHIFTD][SHIFTO][SHIFTS].[6SPC]
[PUR]"
6110 INPUT "LADO[SPC]´A´,´B´[SPC]Y[SPC]
´C´[SPC]";A,B,C
6120 COA=(-A↑2+B↑2+C↑2)/(2*B*C)
6130 AA=-ATN(COA/SQR(-COA*COA+1))+π/
2:A1=AA*180/π
6140 SENB=SIN(AA)*B/A:BB=ATN(SENB/SQ
R(-SENB*SENB+1)):B1=BB*180/π
6150 C1=180-A1-B1:CC=C1*π/180
6160 GOSUB 9000
6170 IF SW=1 THEN GOSUB 10000
6180 GET A$:IF A$<>"R"THEN 6180
6190 RETURN
7000 REM **** SALIDA. ****
7010 PRINT"[CLR][HOM][WHT][RVSON][8SPC]
[SHIFTS][SHIFTA][SHIFTL][SHIFTI][SHIFTD]
[SHIFTA][8SPC]"
7020 PRINT "[PUR][4CRSRD][SPC][SHIFTF]
[SHIFTI][SHIFTN]";:INPUT SN$
7030 IF SN$="SI"THEN END
7040 PRINT "[CR8RD][SPC][SHIFTS]ALID
A[SPC]IMPRESORA";:INPUTSN$
7050 IF SN$="SI"THEN SW=1:GOTO 7070
7060 SW=0
7070 RETURN
9000 REM **** PANTALLA ****
9010 PRINT "[CLR][SPC]LOS[SPC]ANGULO
S[2SPC]´[SHIFTA]´,´[SHIFTB]´[SPC]Y[SPC]
´[SHIFTC]´[SPC]SON:"
9020 PRINT "[SHIFTA]="A1:PRINT "[SHIFTB]
="B1:PRINT "[SHIFTC]="C1
9030 PRINT "LOS[SPC]LADOS[SPC]´A´,´B
´,[SPC]Y[2SPC]´C´[SPC]SON:"
9040 PRINT "A="A:PRINT"B="B:PRINT"C=
"C
9050 PRINT "[2CRSRD][RVSON][WHT][SPC]
PULSA[SPC]LA[SPC]´R´[SPC][PUR]"
9060 RETURN
10000 REM **** IMPRESORA ****
10010 CLOSE4:OPEN4,4,7:PRINT#4
10020 PRINT#4,"[RVSON][19SPC][SHIFTT]
[SPC][SHIFTR][SPC][SHIFTI][SPC][SHIFTA]
[SPC][SHIFTN][SPC][SHIFTG][SPC][SHIFTU]
[SPC][SHIFTL][SPC][SHIFTO][SPC][SHIFTS]
[SPC].[28SPC]"
10030 PRINT#4:PRINT#4
10040 PRINT#4,"[SHIFTL]OS[SPC]ANGULO
S[SPC]VALEN[SPC]EN[SPC]GRADOS:"
10042 PRINT#4:PRINT#4
10050 PRINT#4,"[SHIFTA]="A1:PRINT#4,
"[SHIFTB]="B1:PRINT#4,"[SHIFTC]="C1:
PRINT#4,"[SHIFTA]+[SHIFTB]+[SHIFTC]=
"A1+B1+C1
10060 PRINT#4:PRINT#4,"[SHIFTY][SPC]
EN[SPC]RADIANES[SPC]VALEN:"
10065 PRINT#4:PRINT#4
10070 PRINT#4,"[SHIFTA]="AA:PRINT#4,
"[SHIFTB]="BB:PRINT#4,"[SHIFTC]="CC:
PRINT#4,"[SHIFTA]+[SHIFTB]+[SHIFTC]=
"AA+BB+CC
10080 PRINT#4,"====================
======================================
============="
10090 PRINT#4,"[SHIFTL]OS[SPC]LADOS[SPC]
VALEN:"
10100 PRINT#4,"A="A"[SPC]UNIDADES."
10110 PRINT#4,"B="B"[SPC]UNIDADES."
10120 PRINT#4,"C="C"[SPC]UNIDADES."
10130 PRINT#4
10140 PRINT#4,"[70COMMC]"
10150 PRINT#4:PRINT#4,"[SHIFTL]AS[SPC]
FUNCIONES[SPC]TRIGONOMETRICAS[SPC]DE
[SPC]ESOS[SPC]ANGULOS[SPC]SON:"
```

```
10160 PRINT#4:PRINT#4,"[SHIFTS][SHIFTE]
[SHIFTN][SHIFTO][SPC]DE[SPC]´[SHIFTA]
´="SIN(AA):PRINT#4,"[SHIFTC][SHIFTO]
[SHIFTS][SHIFTE][SHIFTN][SHIFTO][SPC]
DE[SPC]´[SHIFTA]´="COS(AA)
10170 IF COS(AA)=<0 THEN PRINT#4,"[SHIFTT]
[SHIFTA][SHIFTN][SHIFTG][SHIFTE][SHIFTN]
[SHIFTT][SHIFTE][SPC]DE[SPC]´[SHIFTA]
´=[SPC]@[SPC]":GOTO 10172
10171 PRINT#4,"[SHIFTT][SHIFTA][SHIFTN]
[SHIFTG][SHIFTE][SHIFTN][SHIFTT][SHIFTE]
[SPC][SHIFTD][SHIFTE][SPC]´[SHIFTA]´
="TAN(AA)
10172 PRINT#4:PRINT#4,"[SHIFTC][SHIFTO]
[SHIFTS][SHIFTE][SHIFTC][SHIFTA][SHIFTN]
[SHIFTT][SHIFTE][SPC]DE[SPC]´[SHIFTA]
´="1/SIN(AA)
10180 PRINT#4,"[SHIFTS][SHIFTE][SHIFTC]
[SHIFTA][SHIFTN][SHIFTT][SHIFTE][SPC]
DE[SPC]´[SHIFTA]´="1/COS(AA):PRINT#4
,"[SHIFTC][SHIFTO][SHIFTT][SHIFTA][SHIFTN]
[SHIFTG][SHIFTE][SHIFTN][SHIFTT][SHIFTE]
[SPC]DE[SPC]´[SHIFTA]´="1/TAN(AA)
10200 PRINT#4:PRINT#4,"[SHIFTS][SHIFTE]
[SHIFTN][SHIFTO][SPC]DE[SPC]´[SHIFTB]
´="SIN(BB):PRINT#4,"[SHIFTC][SHIFTO]
[SHIFTS][SHIFTE][SHIFTN][SHIFTO][SPC]
DE[SPC]´[SHIFTB]´="COS(BB)
10210 IF COS(BB)<=0 THEN PRINT#4,"[SHIFTT]
[SHIFTA][SHIFTN][SHIFTG][SHIFTE][SHIFTN]
[SHIFTT][SHIFTE][SPC]DE[SPC]´[SHIFTB]
´=[SPC]@[SPC]":GOTO 10212
10211 PRINT#4,"[SHIFTT][SHIFTA][SHIFTN]
[SHIFTG][SHIFTE][SHIFTN][SHIFTT][SHIFTE]
[SPC]DE[SPC]´[SHIFTB]´="TAN(BB)
10212 PRINT#4:PRINT#4,"[SHIFTC][SHIFTO]
[SHIFTS][SHIFTE][SHIFTC][SHIFTA][SHIFTN]
[SHIFTT][SHIFTE][SPC][SHIFTD][SHIFTE]
[SPC]´[SHIFTB]´="1/SIN(BB)
10220 PRINT#4,"[SHIFTS][SHIFTE][SHIFTC]
[SHIFTA][SHIFTN][SHIFTT][SHIFTE][SPC]
DE[SPC]´[SHIFTB]´="1/COS(BB):PRINT#4
,"[SHIFTC][SHIFTO][SHIFTT][SHIFTA][SHIFTN]
[SHIFTG][SHIFTE][SHIFTN][SHIFTT][SHIFTE]
[SPC]DE[SPC]´[SHIFTB]´="1/TAN(BB)
10300 PRINT#4:PRINT#4,"[SHIFTS][SHIFTE]
[SHIFTN][SHIFTO][SPC]DE[SPC]´[SHIFTC]
´="SIN(CC):PRINT#4,"[SHIFTC][SHIFTO]
[SHIFTS][SHIFTE][SHIFTN][SHIFTO][SPC]
DE[SPC]´[SHIFTC]´="COS(CC)
10310 IF COS(CC)<=0 THEN PRINT#4,"[SHIFTT]
[SHIFTA][SHIFTN][SHIFTG][SHIFTE][SHIFTN]
[SHIFTT][SHIFTE][SPC]DE[SPC]´[SHIFTC]
´=[SPC]@[SPC]":GOTO 10312
10311 PRINT#4,"[SHIFTT][SHIFTA][SHIFTN]
[SHIFTG][SHIFTE][SHIFTN][SHIFTT][SHIFTE]
[SPC]´[SHIFTC]´="TAN(CC)
10312 PRINT#4:PRINT#4,"[SHIFTC][SHIFTO]
[SHIFTS][SHIFTE][SHIFTC][SHIFTA][SHIFTN]
[SHIFTT][SHIFTE][SPC]DE[SPC]´[SHIFTC]
´="1/SIN(CC)
10320 IF COS(CC)<=0THEN PRINT#4,"[SHIFTS]
[SHIFTE][SHIFTC][SHIFTA][SHIFTN][SHIFTT]
[SHIFTE][SPC]DE[SPC]´[SHIFTC]´=[SPC]
@[SPC]":PRINT#4,"[SHIFTC][SHIFTO][SHIFTT]
[SHIFTA][SHIFTN][SHIFTG][SHIFTE][SHIFTN]
[SHIFTT][SHIFTE][SPC]DE[SPC]´[SHIFTC]
´=[SPC]@[SPC]":GOTO 10330
10321 PRINT#4,"[SHIFTS][SHIFTE][SHIFTC]
[SHIFTA][SHIFTN][SHIFTT][SHIFTE][SPC]
DE[SPC]´[SHIFTC]´="1/COS(CC):PRINT#4
,"[SHIFTC][SHIFTO][SHIFTT][SHIFTA][SHIFTN]
[SHIFTG][SHIFTE][SHIFTN][SHIFTT][SHIFTE]
[SPC]DE[SPC]´[SHIFTC]´="1/TAN(CC)
10330 REM *
11140 PRINT#4,"*********************
****************************************
*************"
11150 RETURN
```

Hundir la flota

Luis Marcelo García Solana
Cristo de los Remedios, 22, 1.ª
S. Sebastián de los Reyes (Madrid)

C-64



Hundir la flota consiste en destruir un barco mediante los torpedos lanzados por un submarino; el barco navega desde el mar hasta la entrada del puerto o viceversa; su trayectoria no es continua, es decir, el barco puede salir desde el puerto, llegar a la parte izquierda de la pantalla y volver a salir otra vez del puerto, o darse la vuelta, etc. (Su misión es despistar al jugador para que no sepa por dónde sale el barco).
Al mismo tiempo el barco irá disparando cargas de profundidad. El juego a "primera vista" parece muy simple, pero este se va acomodando a la habilidad del jugador. Pueden existir casos en que al disparar, el barco detecte la posicióndel submarino y dé la vuelta, no es extrañéis, está diseñado así. El juego está animado con efectos sonoros y visuales.

```
1 REM HUNDIR LA FLOTA
2 S=54272:FORL=1TO24:POKES+L,0:NEXT
3 POKES+24,4:POKES+5,15:POKES+6,200
4 PRINT"[BLU][CLR]"
5 POKE53281,5:POKE53280,0
6 S=54272:FORL=1TO24:POKES+L,0:NEXT
8 POKES+24,4:POKES+5,15:POKES+6,200
9 FORYT=23TO1STEP-1:PRINTTAB(YT)"[RED]
HUNDIR[SPC]LA[SPC]FLOTA":POKES+4,33:
POKES+1,YT:NEXT
10 PRINT"[24CRSRU]"
11 FOR E1=1TO23:PRINTTAB(E1)"[BLU]HU
NDIR[SPC]LA[SPC]FLOTA":POKES+4,33 :P
OKES+1,E1:NEXT
13 PRINT"[2CRSRU][3CRSRR][10CRSRU][RED]
POR[2CRSRU]":FORT=10TO1STEP-.4:POKES
+24,T:POKES+4,129
14 POKES+1,5:NEXT: PRINT"[32CRSRR][RED]
L.M.G."
15 PRINT"[CRSRD][32CRSRR][RED]SOLANA
"
16 FORT=10TO1STEP-.4:POKES+24,T: POK
ES+4,129:POKES+1,5:NEXT
18 IFQE=1THEN POKES+24,8:GOTO28
19 POKES+24,8:POKES+4,33:FORTY=1TO36
:READA:READB
20 READC:FORT=1TOC/3:POKES+1,A:NEXT:
NEXT:GOTO29
```

```
21 DATA 28,214,90 ,19,63,90,28,214,9
0,19,63,90 ,28,214,90 ,19,63,90
22 DATA28,214,90 ,25,177,55,22,227,5
5,21,154,55,22,227,90
23 DATA15,70,90 ,22,227,100,15,70,10
0, 22,227,75,15,70,100,22,227,100,21
,154,75
24 DATA19,63,75,17,37,100,25,177,100
,17,37,100
25 DATA25,177,100,17,37,100,25,177,1
00,17,37,100,22,227,75,21,154,75
26 DATA19,63,75,17,37,75,16,47,100,1
7,37,100,19,63,100,20,100,150,19,63,
100
27 DATA 17,337,150
28 FORR=1TO20: FORT=1TO200STEP10:POK
ES+4,17:POKES+1,T:NEXT:NEXT:POKES+4,
0
29 PRINT"[CLR]":GOTO3000
30 PRINT"[CLR][7CRSRD]"
34 FORA=1TO15:PRINT"[BLU][RVSON][40SPC]
";:NEXT
35 FORA=920TO1000:POKE1024+A,224:POK
E55296+A,9:NEXT
36 FORA=354TO359:POKE1024+A,224:POKE
55296+A,0 :NEXT
37 FORA=314TO319:POKE1024+A,114:POKE
55296+A,0:NEXT
50 V=53248
60 POKEV+21,38
70 POKEV+40,10:POKEV+41,0:POKEV+42,1
0
80 POKE2041,193:POKE2042,194:POKE204
3,195:POKE2044,196:POKE2045,197
90 POKEV+2,63:POKEV+6,67
92 POKEV+29,54:POKEV+4,0:POKEV+23,8
100 POKEV+3,198:POKEV+5,98 :POKEV+7,
205
109 IFQE=1 THEN218
110 FORX=0TO62:READA:POKE12352+X,A:N
EXT
120 FORX=0TO62:READB:POKE12416+X,B:N
EXT
125 FORX=0TO62:READC:POKE12480+X,C:N
EXT
126 FORX=0TO62:READR:POKE12544+X,R:N
EXT
127 FORX=0TO62:READD:POKE12608+X,D:N
EXT
128 FORX=0TO62:READF:POKE12800+X,F:N
EXT
129 FORX=0TO62:READP:POKE12864+X,P:N
EXT
130 FORX=0TO62:READZ:POKE12928+X,Z:N
EXT
135 FORX=0TO62:READL:POKE12992+X,L:N
EXT
140 POKES+4,0
218 01=4:PP=4:TI$="000000"
219 POKEV+44,0
220 POKES+4,0:POKEV+7,205 :POKES+24,
4:POKES+5,15:POKES+6,200
221 IFP1>2THEN01=6
222 IFP1>4THEN01=8:PP=6
223 IFP1>6 THEN01=10:PP=7
224 IFP1>8 THEN01=12:PP=8
225 IFP1>10 THEN01=10:PP=7
226 IF P1>12 THEN01=12:PP=8
228 IFB=0ORB=254 THENGOTO4000
230 IFAU=3ORAU=1THENB=B-01:GOTO238
235 IFAU=4ORAU=2THENB=B+01
238 IFB<0ORB>254 THENGOTO4000
258 IFPEEK(197)=10THENX=X-PP
259 IFPEEK(197)=13THENX=X+PP
260 IFX>255THENX=255
270 IFX<30THENX=30
271 POKEV+10,B:G=G+5:IFG>160THENG=0
272 POKEV+11,92+G
273 IFPEEK(V+30)ANDG=105ANDB>X-50AND
B<X+50  THEN 1000
280 POKEV+2,X:POKEV+6,X:POKEV+4,B
290 IFPEEK(197)=54THENPOKEV+21,46:P2
=P2+1:GOTO294
293 GOTO220
294 FORE=1TO154STEP4
295 POKES+4,129:POKES+1,15+E:POKES,7
0
300 C=205-E
304 IFE>97  THENPOKEV+42,6
305 B=B+01
306 IFB>254THEN4000
310 POKEV+7,C:POKEV+4,B
330 IFPEEK(V+30)ANDC>103ANDC<117ANDG
<>0THEN500
331 POKEV+10,B:G=G+5
332 IFG>160 THENG=0
333 POKEV+11,92+G
334 IFPEEK(V+30)ANDG=105ANDB>X-50AND
B<X+50  THEN 1000
340 IFPEEK(197)=10THENX=X-PP
350 IFPEEK(197)=13THENX=X+PP
360 IFX>255THENX=255
370 IFX<30 THENX=30
380 POKEV+2,X
390 NEXT
400 IFPEEK(197)=54THEN294:P2=P2+1
405 POKEV+21,38 :POKEV+42,8
410 GOTO220
500 POKEV+43,0:POKEV+8,B:POKEV+9,98:
POKEV+21,18:P=0
508 POKES+24,15:FORW=15TO0STEP-.2:PO
KES+24,W:POKES+4,129:POKES+5,15:POKE
S+6,200
509 POKES+1,5:POKES,25:P=P+1:POKEV+9
,98+P:NEXT
570 FORT=175TO215STEP.1:POKEV+9,T:NE
XT:FORZ=1TO300:NEXT
580 POKEV+21,38:B=0:X=95:POKEV+2,X:P
1=P1+1:POKEV+42,8:GOTO4000
830 DATA0,0,0,0,8,0,0,8,0,0,8,0,0,24
,0,0,24,0,0,24,0,0,56,0,0,56,0,0,56,
0
840 DATA63,255,248,127,255,252,255,2
55,254,127,255,252,63,255,248
850 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0
860 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,16,0,0,16,0,0,16,0
870 DATA0,24,0,0,24,0,0,255,0,0,255,
0,243,255,207,63,255,254
880 DATA31,255,252,15,255,248,7,255,
240,3,255,224,0,0,0
890 DATA0,0,0,8,0,0,8,0,0,28,0,0,2
8,0,0,0,0,0,0 ,0,0,0,0,0,0,0,0,0
900 DATA0,0,0 ,0,0,0 ,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0
910 DATA0,0,0,0,0,0,0,0,0
920 DATA 0,0,0,34,3,32,0,7,0,0,0,0,0
,0,0,56,0,14,60,128,94,24,195,124
930 DATA4,127,127,0,63,254,7,31,254,
32,15,249,6,71,250,18,7,246,0,111,23
3
940 DATA0,31,222,24,63,190,0,127,128
,0,126,128,0,254,0,0,0,0
950 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0
960 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,56,0,14,56,0,14,56,0,14
965 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,4,209,128,6,153,
128
966 DATA0,0,0,1,205,128,99,255,128,4
7,255,216,15,55,0,0,0,0
967 DATA3,51,128,2,1,0,0,0,0,0,0,0,0
,0,0
968 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,
```

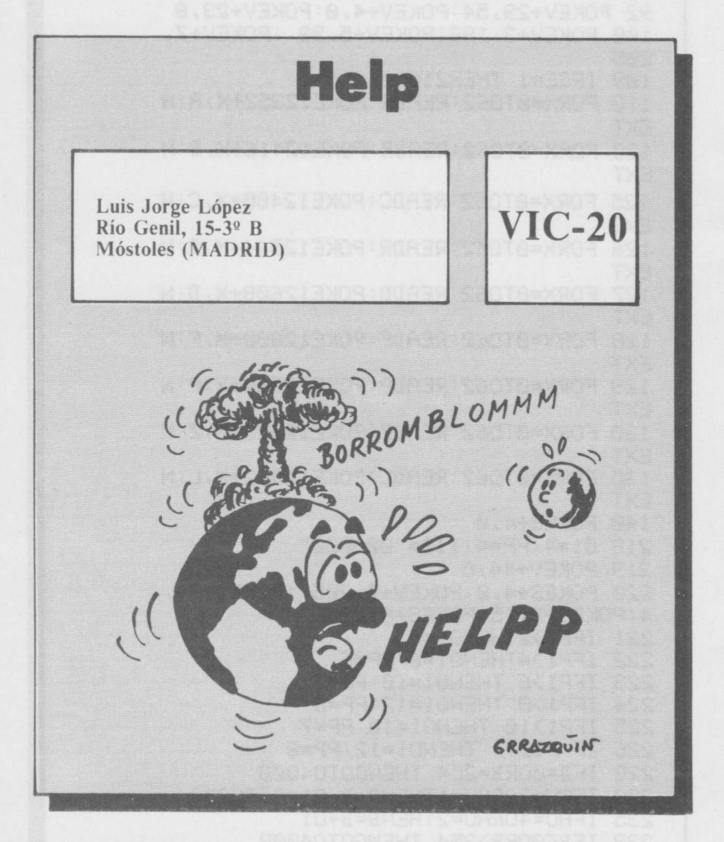
```
0,17,144,0,25,25,128,0,1,128,0,0,0,8
,0,0,24
969 DATA230,192,1,238,192,0,17,224,1
95,121,0,79,53,0,15,57,24
970 DATA 0,0,03,53,0,4,2,0,0,0,0,0,0
,0,0
975 DATA0,0,0,0,0,0,0,0,0,35,16,0,50
,24,96,0,0,96,0,0,0,0,0,0,16,227
980 DATA 0,49,251,48,0,0,48,0,0,120,
0,0,0,0,16,0,140,120
990 DATA 64,152,48,192,28,49,192,0,0
,6,0,0,0,1,152,224,1,0,64
992 DATA70,8,0,98,12,48,0,0,48,0,0,0
,0,0,0,0,0,0,0,0,0,16,109
994 DATA192,48,241,192,0,0,0,0,0,0,0
,0,6,0,0,6,0,0,15,0
996 DATA0,0,0,0,0,0,0,0,0,16,0,0,120
,0,6,48,64,30,48,193,31,1,192
1000 POKEV+21,6:POKES+5,15:POKES+6,8
0
1004 FORG=1TO10 :A=A+20:H=H+1:IFA>10
0 THEN A=0
1005 IFH=15 THEN H=0
1006 POKEV+40,H:POKES+4,33:FORW=102-
ATO1STEP-6
1010 POKES+1,W:POKES,70
1030 NEXT:NEXT:H=0:POKEV+23,2
1040 POKE2041,200:FORT=15TO12STEP-.8
:POKES+24,T:POKES+4,129:POKES+5,15
1045 POKES+6,200:POKES+1,3:POKES,25:
NEXT:POKE2041,201:FORT=12TO8STEP-.8
1050 POKES+24,T:POKES+4,129:POKES+5,
15:POKES+6,200:POKES+1,3:POKES,25:NE
XT
1055 POKE2041,202:FORT=8TO5STEP-.8:P
OKES+24,T:POKES+4,129:POKES+5,15
1060 POKES+6,200:POKES+1,3:POKES,25:
NEXT:POKE2041,203:FORT=5TO0STEP-.09
1065 POKES+24,T:POKES+4,129:POKES+5,
15:POKES+6,200:POKES+1,3:POKES,25:NE
XT
1070 POKEV+21,4:POKEV+23,8:POKE2041,
193
1080 POKEV+40,10:POKEV+21,46:X=100:P
OKEV+2,X :POKEV+4,0:B=0:POKEV+11,92
1090 P3=P3+1:G=0:POKEV+42,8:IFP3>1TH
EN 5000
2000 GOTO220
3000 PRINT"[CRSRR][BLU][11CRSRR][15COMMP]
"
3001 PRINT"[11CRSRR][SHIFTN][14SPC][SHIFTN]
[COMMH]"
3005 PRINT"[11CRSRR][SHIFTO][13COMMY]
[SHIFTP][SPC][COMMH]"
3020 PRINT"[11CRSRR][COMMH]INSTRUCCI
ONES[COMMN][SPC][COMMH]"
3030 PRINT"[BLU][11CRSRR][SHIFTL][13COMMP]
[SHIFT@][SHIFTN]"
3031 PRINT"[YEL][2CRSRD][8CRSRR]PULS
AR[SPC][RED]A[BLU][SPC]PARA[SPC]IZQU
IERDA"
3032 PRINT"[8CRSRR]PULSAR[SPC][RED]S
[BLK][SPC]PARA[SPC]DERECHA"
3033 PRINT"[8CRSRR]PULSAR[SPC][RED]↑
[YEL][SPC]PARA[SPC]DISPARO"
3040 PRINT"[2CRSRR][COMM2][CRSRD][4CRSRR]
.......D[SPC]A[SPC]T[SPC]O[SPC]S...
..[RED]PUNTOS"
3045 PRINT"[2CRSRR][WHT][CRSRD][4CRSRR]
POR[SPC]BARCO[2SPC]HUNDIDO[SPC][RED]
[CRSRR][4SPC]250[SPC]"
3050 PRINT"[2CRSRR][BLU][CRSRD][4CRSRR]
POR[SPC]TORPEDO[SPC]DISPARADO[RED][3SPC]
-13"
3070 PRINT"[CRSRR][BLK][CRSRD][SPC][4CRSRR]
DISPONES[SPC]DE[SPC][RED]2[BLK][SPC]
SUBMARINOS"
3080 PRINT"[3CRSRD][RVSON][RED][2CRSRR]
PRESIONA[2SPC]UNA[2SPC]TECLA[2SPC]PA
RA[2SPC]EMPEZAR"
```

```
3090 FORP=1TO25 STEP.9:POKES+4,33:PO
KES+1,P:NEXT
3095 IFPEEK(197)=64 THEN3090
3100 :POKES+4,0:GOTO30
4000 AU=INT(RND(1)*4)+1:IFAU=3ORAU=1
THENB=254
4010 IFAU=2ORAU=4THENB=0
4020 POKEV+42,8:GOTO 230
5000 PRINT"[CLR]":POKE53281,6:POKE53
280,7:POKEV+21,0
5005 PRINT"[RVSOFF][5CRSRD][YEL][3CRSRR]
TU[SPC]TIEMPO[SPC]ES[SPC]DE[2SPC]";T
I$"
5006 PRINT"[CRSRR][CRSRU][27CRSRR]SE
GUNDOS"
5008 TT=P1*250+P2*-13
5009 PRINT"[CRSRR][3CRSRD][9CRSRR]PU
NTUACION[SPC]TOTAL";TT
5010 PRINT"[2CRSRD][10CRSRR]FIN[3SPC]
DE[3SPC]PARTIDA"
6000 PRINT"[2CRSRD][RVSON][10CRSRR][SPC]
ESPERA[SPC]UN[SPC]MOMENTO[SPC]SI"
6004 PRINT"[2CRSRD][RVSON][9CRSRR][SPC]
QUIERES[SPC]JUGAR[SPC]DE[SPC]NUEVO"
:FORL=0TO24:POKES+L,0:NEXT
6005 FORL=1TO73:AO=AO+6:POKES+24,8:P
OKES+5,15:POKES+6,200:IF AO>150 THEN
AO=0
6007 FORLP=100+AOTO10STEP-20:POKES+4
,33:POKES+1,LP:NEXT:NEXT
6010 P1=0:P2=0:P3=0:QE=1:PL=100:GOTO
4
```

Help

Luis Jorge López
Río Genil, 15-3º B
Móstoles (MADRID)

VIC-20



Este programa le he puesto yo como **ayuda escolar**, pero sirve como diario, agenda, guía telefónica, etc.
El programa no es de un estilo muy bueno, pero son mis primeros pasos en la programación.
Es para el VIC-20 con o sin ampliaciones.
Espero que les interese, puesto que puede serle útil a alguien.
El programa consiste en la apertura de una serie de ficheros en el datassette, que luego pueden ser leídos otra vez por el ordenador.

```
10 POKE 36879,110
20 PRINT"[CLR]"
25 PRINT"[WHT]"
30 X$="[2SPC][COMM+][SPC][COMM+][2SPC]
[3COMM+][2SPC][COMM+][4SPC][3COMM+][2SPC]
"
40 FOR I=1 TO 22:POKE36874,245:FORN=
15TO0STEP-1:POKE36878,N:NEXTN
50 POKE36874,0
60 PRINTMID$(X$,I,1);:NEXTI
70 J$="[2SPC][COMM+][SPC][COMM+][2SPC]
[COMM+][4SPC][COMM+][4SPC][COMM+][SPC]
[COMM+][2SPC]"
80 FOR I=1TO 22:POKE36875,155:FORN=1
5TO0STEP-1:POKE36878,N:NEXTN
90 POKE36875,0
100 PRINTMID$(J$,I,1);:NEXTI
110 B$="[2SPC][3COMM+][2SPC][2COMM+]
[3SPC][COMM+][4SPC][3COMM+][2SPC]"
120 FORI=1TO22:POKE36876,176:FORN=15
TO0STEP-1:POKE36878,N:NEXTN
130 POKE36876,0
140 PRINTMID$(B$,I,1);:NEXTI
150 G$="[2SPC][COMM+][SPC][COMM+][2SPC]
[COMM+][4SPC][COMM+][4SPC][COMM+][4SPC]
"
160 FOR I=1TO22:POKE36877,255:FORN=1
5TO0STEP-1:POKE36878,N:NEXTN
170 POKE36877,0
180 PRINTMID$(G$,I,1);:NEXTI
190 H$="[2SPC][COMM+][SPC][COMM+][2SPC]
[3COMM+][2SPC][3COMM+][2SPC][COMM+][4SPC]
"
200 FORI=1TO22:POKE36876,198:FORN=15
TO0STEP-1:POKE36878,N:NEXTN
210 POKE36876,0
220 PRINTMID$(H$,I,1);:NEXTI
225 PRINT"[3CRSRD]"
230 L$="[SPC](C)[SPC]LUIS[SPC]JORGE[SPC]
LOPEZ[2SPC]"
240 FOR I=1TO22:FORJ=1TO150:NEXTJ
250 PRINTMID$(L$,I,1);:NEXTI
260 FOR L=1TO 200:NEXTL
270 PRINT"[CLR]"
280 GOSUB 50000
320 PRINT"[3CRSRD]"
330 PRINT"ESTE[SPC]PROGRAMA[SPC]TE[SPC]
SERVIRA[SPC]COMO[SPC]AYUDA(HELP)EN[SPC]
LOS[SPC]ESTUDIOS."
340 PRINT"CON[SPC]EL[SPC]PODRAS[SPC]
REPASAR[SPC]TUS[SPC]APUNTES[SPC]O[SPC]
RESUMIR"
350 PRINT"LAS[SPC]PARTES[SPC]MAS[SPC]
IMPORTANTES[SPC]DE[SPC]UNA[SPC]LECCI
ON."
360 FOR N=1 TO 8000:NEXTN
370 PRINT"[CLR]"
380 GOSUB 50000
420 PRINT"[3CRSRD]"
430 PRINT"PARA[SPC]ELLO[SPC]DEBERAS[SPC]
INTRODUCIR[SPC]TUS[SPC]APUNTES[SPC]O
[SPC]RESUMENES[SPC]DEL[SPC]SIGUIENTE
[SPC]MODO."
440 PRINT"ELEGIRAS[SPC]ENTRE[SPC]REP
ASO[SPC]O[SPC]INTRODUCIR[SPC]."
450 FOR N=1TO8000:NEXTN
460 PRINT"[CLR]"
470 GOSUB 50000
480 PRINT"[3CRSRD]"
490 PRINT"PARA[SPC]REPASAR[SPC]PULSA
[2SPC]R[SPC]Y[SPC]ESPERA"
500 PRINT"PARA[SPC]INTRODUCIR[SPC]PU
LSA[SPC]I[SPC]Y[SPC]HAZ[SPC]LO[SPC]Q
UE[SPC]TE[SPC]DIGA[SPC]EL[SPC]ORDENA
DOR"
505 FOR N=1 TO 8000:NEXTN
510 PRINT"[CLR]"
511 GOSUB 50000
512 PRINT"[3CRSRD]"
513 PRINT"SI[SPC]DECIDES[SPC]REPASAR
[SPC]DEBERAS,DESPUES[SPC]DE[SPC]HABE
R[SPC]PULSADO[SPC]LA[SPC]R"
514 PRINT"[SPC]INTRODUCIR[SPC]EL[SPC]
NOMBRE[SPC]Y[SPC]EL[SPC]NO.QUE[SPC]D
ISTE[SPC]A[SPC]TUS[SPC]APUNTES[SPC]C
UANDO[SPC]LOS[SPC]INTRODUCISTE"
515 PRINT"DEBERAS[SPC]TENER[SPC]PREP
ARADA[SPC]LA[SPC]CINTA"
520 FOR J=1TO8000:NEXTJ
530 PRINT"[CLR]"
532 GOSUB 50000
534 PRINT"[3CRSRD]SI[SPC]DECIDES[SPC]
INTRODUCIR[SPC]DEBERAS[SPC]METER[SPC]
EL[SPC]TEXTO[SPC]EN[SPC]PARRAFOS[SPC]
NO[SPC]MAYOR[SPC]DE[SPC]4[SPC]LI"
536 PRINT"NEAS[SPC]ACTO[SPC]SEGUIDO[SPC]
PULSA[SPC]RETURN[SPC]Y[SPC]CONTINUA[SPC]
ASI[SPC]SIN[SPC]EXCEDERTE[SPC]EN[SPC]
LA[SPC]CANTIDAD"
538 PRINT"DEL[SPC]TEXTO[SPC]QUE[SPC]
INTRODUCES"
790 PRINT"PULSA[SPC]I[SPC]O[SPC]R"
800 GET A$:IF A$=""THEN 800
810 IF A$="R"THENGOTO2000
820 IF A$="I"THEN GOTO10000
1000 GOTO800
2000 PRINT"[CLR]"
2005 GOSUB 50000
2010 PRINT"[2CRSRD]NUMERO[SPC]DE[SPC]
FICHERO[SPC]QUE[SPC]DISTE":INPUTA
2020 PRINT"NOMBRE[SPC]QUE[SPC]ASIGNA
STE":INPUTZ$
2030 OPENA,1,0,Z$
2035 PRINT"[2CRSRR][RVSON]FICHERO[SPC]
ABIERTO[RVSOFF]"
2039 INPUT#A,A$
2040 PRINTA$:GOSUB60500
2041 INPUT#A,S$
2042 PRINTS$:GOSUB60500
2043 INPUT#A,D$
2044 PRINTD$:GOTO60500
2045 INPUT#A,F$
2046 PRINTF$:GOSUB60500
2047 INPUT#A,G$
2048 PRINTG$:GOSUB60500
2050 GOSUB 60000
2051 INPUT#A,H$
2052 PRINTH$:GOSUB60500
2053 INPUT#A,J$
2054 PRINTJ$:GOSUB60500
2055 INPUT#A,K$
2056 PRINTK$:GOSUB60500
2057 INPUT#A,L$
2058 PRINTL$:GOSUB60500
2059 INPUT#A,Q$
2060 PRINTQ$:GOSUB60500
2061 INPUT#A,W$
2062 PRINTW$:GOSUB60500
2063 INPUT#A,E$
2064 PRINTE$:GOSUB60500
2066 GOSUB 60000
2067 INPUT#A,R$
2068 PRINTR$:GOSUB60500
2069 INPUT#A,T$
2070 PRINTT$:GOSUB60500
2071 INPUT#A,Y$
2072 PRINTY$:GOSUB60500
2073 INPUT#A,U$
2074 PRINTU$:GOSUB60500
2075 INPUT#A,I$
2076 PRINTI$:GOSUB60500
2077 INPUT#A,O$
2078 PRINTO$:GOSUB60500
2079 INPUT#A,P$
2080 PRINTP$:GOSUB60500
2081 INPUT#A,V$
2082 PRINTV$:GOSUB60500
2083 PRINT"PULSA[SPC]UNA[SPC]TECLA"
```

```
2100 GET Z$:IF Z$=""THEN 2100
2110 PRINT"[CLR]":GOSUB 50000
2120 GOTO 20500
10000 PRINT"[CLR]"
10010 GOSUB 50000
10020 PRINT"[CRSRD]AHORA[SPC]INTRODU
CIRAS[SPC]LOS[SPC]DATOS[SPC]EN[SPC]P
ARRAFOS[SPC]DE[SPC]4[SPC]LINEAS[SPC]
MAXIMO[SPC]Y[SPC]SOLO[SPC]20[SPC]PAR
RAFOS"
10030 PRINT"POR[SPC]FICHERO,O[SPC]SE
A[SPC]80[SPC]LINEAS.CUANDO[SPC]YA[SPC]
NO[SPC]QUIERAS[SPC]METER[SPC]MAS[SPC]
DATOS[SPC]Y[SPC]EL[SPC]ORDENADOR"
10040 PRINT"TE[SPC]LOS[SPC]PIDA[SPC]
PON[SPC]STOP[SPC]Y[SPC]DA[SPC]AL[SPC]
RETURN[SPC]ASI[SPC]HASTA[SPC]QUE[SPC]
NO[SPC]TE[SPC]PIDA"
10050 PRINT"HAZ[SPC]LO[SPC]QUE[SPC]T
E[SPC]INDIQUE[SPC]EL[SPC]ORDENADOR,[SPC]
PASO[SPC]A[SPC]PASO"
10060 FOR N=1 TO15000:NEXT N
10070 PRINT"[CLR][CRSRD]"
10080 GOSUB 50000
10090 PRINT"DEBERAS[SPC]ASIGNAR[SPC]
UN[SPC]NUMERO[SPC]DIFERENTE[SPC]A[SPC]
CADA[SPC]FICHERO[SPC]PERO[SPC]SEGUID
OS[SPC]EJ:1,2,3,4"
10100 PRINT"Y[SPC]TAMBIEN[SPC]DARLE[SPC]
UN[SPC]NOMBRE[SPC]POR[SPC]EJEMPLO[SPC]
FISICA[SPC]1[SPC]O[SPC]COMO[SPC]PREF
IERAS[SPC]PERO[SPC]DEBERAS"
10110 PRINT"APUNTAR[SPC]TANTO[SPC]EL
[SPC]NOMBRE[SPC]COMO[SPC]EL[SPC]NUME
RO[SPC]PARA[SPC]CUANDO[SPC]QUIERAS[SPC]
REPASARLO"
10115 PRINT"SI[SPC]EL[SPC]ORDENADOR[SPC]
AL[SPC]METER[SPC]ALGUNOS[SPC]DATOS[SPC]
TE[SPC]DICE[SPC]EXTRA[SPC]IGNORED[SPC]
VUELVE[SPC]A[SPC]METER[SPC]LOS"
10116 PRINT"MISMOS[SPC]DATOS,[SPC]PO
R[SPC]ULTIMO[SPC]MANTEN[SPC]EL[SPC]D
ATASSETTE[SPC]CON[SPC]RECORD[SPC]&[SPC]
PLAY[SPC]PULSADO"
10120 FOR N=1 TO20000:NEXTN
10130 PRINT"[CLR]":GOSUB 50000
10140 PRINT"[CRSRD]DA[SPC]NUMERO":IN
PUT A
10150 PRINT"DA[SPC]NOMBRE":INPUT Z$
10160 OPEN A,1,1,Z$
10170 PRINT"[2CRSRR][RVSON]ESCRIBA[SPC]
DATOS"
10180 PRINT
10190 INPUT A$
10191 PRINT#A,A$:GOSUB60500
10192 INPUT S$
10193 PRINT#A,S$:GOSUB60500
10194 INPUTD$
10195 PRINT#A,D$:GOSUB60500
10196 INPUTF$
10197 PRINT#A,F$:GOSUB60500
10198 INPUTG$
10199 PRINT#A,G$:GOSUB60500
10200 INPUTH$
10201 PRINT#A,H$:GOSUB60500
10202 INPUTJ$
10203 PRINT#A,J$:GOSUB60500
10204 INPUTK$
10205 PRINT#A,K$:GOSUB60500
10206 INPUTL$
10207 PRINT#A,L$:GOSUB60500
10208 INPUTQ$
10209 PRINT#A,Q$:GOSUB60500
10210 INPUTW$
10211 PRINT#A,W$:GOSUB60500
10212 INPUTE$
10213 PRINT#A,E$:GOSUB60500
10214 INPUTR$
10215 PRINT#A,R$:GOSUB60500
10216 INPUTT$
10217 PRINT#A,T$:GOSUB60500
10218 INPUTY$
10219 PRINT#A,Y$:GOSUB60500
10220 INPUTU$
10221 PRINT#A,U$:GOSUB60500
10222 INPUTI$
10223 PRINT#A,I$:GOSUB60500
10224 INPUTO$
10225 PRINT#A,O$:GOSUB60500
10226 INPUTP$
10227 PRINT#A,P$:GOSUB60500
10228 INPUTV$
10229 PRINT#A,V$:GOSUB60500
20000 CLOSE A
20100 PRINT"PULSA[SPC]UNA[SPC]TECLA"

20200 GETZ$:IF Z$=""THEN20200
20300 PRINT"[CLR]"
20400 GOSUB 50000
20500 PRINT"[CRSRD]SI[SPC]QUIERES[SPC]
REPASAR[SPC]O[SPC]INTRODUCIR[SPC]PUL
SA[SPC]F1[SPC]Y[SPC]SI[SPC]NO[SPC]PU
LSA[SPC]F7"
20600 GET N$:IF N$=""THEN20600
20700 IF N$="[F1]" THEN 790
20750 IF N$="[F7]" THEN PRINT"[CLR][RVSON]
ME[SPC]ESTOY[SPC]DESTRUYENDO[RVSOFF]
":FOR N=1TO 1000:NEXTN:SYS 64802
50000 PRINTTAB(4)"VIC-20[3SPC]HELP"
50010 FOR J=0TO21
50020 PRINTCHR$(164);
50025 NEXTJ
50030 RETURN
60000 PRINT"PULSA[SPC]UNA[SPC]TECLA"

60010 GET Z$:IF Z$=""THEN 60010
60020 RETURN
60500 IF A$="STOP"THEN 20000
60510 IF S$="STOP"THEN 20000
60520 IF D$="STOP"THEN 20000
60530 IF F$="STOP"THEN 20000
60540 IF G$="STOP" THEN 20000
60550 IF H$="STOP" THEN 20000
60560 IF J$="STOP" THEN 20000
60570 IF K$="STOP" THEN 20000
60580 IF L$="STOP" THEN 20000
60590 IF Q$="STOP" THEN 20000
60600 IF W$="STOP" THEN 20000
60610 IF E$="STOP" THEN 20000
60620 IF R$="STOP" THEN 20000
60630 IF T$="STOP" THEN 20000
60640 IF Y$="STOP" THEN 20000
60650 IF U$="STOP" THEN 20000
60660 IF I$="STOP" THEN 20000
60670 IF O$="STOP" THEN 20000
60680 IF P$="STOP" THEN 20000
60690 IF V$="STOP" THEN 20000
60700 RETURN
```



Protecciones contra escritura

Las etiquetas de protección de escritura para los discos son ideales para las cintas de cassette.

Después de haber roto la protección de la cinta, tápala con una de estas etiquetas cuando quieras grabar en ella.

L.F.S. RUN EE.UU.

Menú cinta

Tony Rueda Roselló
Padilla, 216
Telf. 2459938 - BARCELONA

C-64



Este programita se graba al principio de cada cinta y después se pueden grabar los demás programas.

Este programa que he hecho en el C-64 también puede cargarse en el VIC-20, sólo teniendo que cambiar los dos POKES de color de la línea 20 del programa (fondo y borde azul claro). Así también puede ir con la unidad de discos siendo mucho más rápido.

El Menú Cinta te da las siguientes opciones:

—Ver los programas contenidos en la cinta.
—Ver el contador de cada programa (para buscarlo más rápidamente en el Datacassette).
—Buscar el programa automáticamente, solamente tocando el número de opción (eficaz en la unidad de discos).
—Ver notas sobre la cinta (o disco) y los programas contenidos en ella.

En el programa he puesto el ejemplo de Video Casino, pudiéndolo cambiar por otro accediendo al listado. A continuación os pongo las partes del listado para cambiar los nombres:

—Líneas 1999-2040, para cambiar la cabecera del programa.
—Líneas 4999-5060, para cambiar el nombre de los programas.
—Líneas 7999-8060, para cambiar el número de los contadores.
—Líneas 9999-10050, para cambiar los nombres del comando Load.
—Líneas 10599-10633, para cambiar las notas.

Espero que os sea de utilidad y sin nada más que añadir se despide un commodoriano.

¡Adeu!

NOTA: Para los usuarios de la Unidad de disco flexible tienen que añadir al comando LOAD el ,8 (que hace acceder a dicha unidad).

```
3 REM ******************************
4 REM *       MENU CINTA           *
5 REM *                            *
6 REM * AUTOR: TONY RUEDA ROSELLO *
7 REM *                            *
8 REM *       JULIO 1984           *
9 REM ******************************
10 :
20 POKE53280,14:POKE53281,14
21 REM ********************
22 REM *   MENU PRINCIPAL   *
23 REM ********************
30 PRINT"[CLR]"
35 PRINTCHR$(14)"[BLU]"
40 GOSUB2000
50 PRINT:PRINT:PRINT"[2SPC][SHIFTM][SHIFTE]
[SHIFTN][SHIFTU][SPC][SHIFTP][SHIFTR]
[SHIFTI][SHIFTN][SHIFTC][SHIFTI][SHIFTP]
[SHIFTA][SHIFTL]:"
60 PRINT:PRINT"[3SPC][SHIFTE]LIJA:"
70 PRINT:PRINT"[2CRSRR][SHIFTA].[CRSRR]
[SHIFTM]IRAR[SPC]LOS[SPC]PROGRAMAS[SPC]
QUE[SPC]CONTIENE[8CRSRR]LA[SPC]CINTA
."
80 PRINT:PRINT"[2CRSRR][SHIFTB].[CRSRR]
[SHIFTM]IRAR[SPC]EL[2SPC]CONTADOR[2SPC]
DE[SPC]CADA[SPC]PRO-[8CRSRR]GRAMA."
90 PRINT:PRINT"[2CRSRR][SHIFTC].[CRSRR]
[SHIFTC]ARGAR[SPC]EL[SPC]PROGRAMA[SPC]
ELEGIDO[SPC]EN[SPC]EL[8CRSRR]ORDENAD
OR."
100 PRINT:PRINT"[2CRSRR][SHIFTD].[CRSRR]
[SHIFTN]OTAS[SPC]DE[SPC]INTERES."
130 GETX$:IFX$=""THEN130
140 IFX$="A"THEN500
150 IFX$="B"THEN1000
160 IFX$="C"THEN1500
170 IFX$="D"THEN1800
180 GOTO130
499 REM ************************
500 REM *   MENU SECUNDARIO (A)   *
501 REM ************************
510 PRINT"[CLR]"
515 GOSUB2000
520 PRINT:PRINT:PRINT"[2CRSRR][SHIFTP]
[SHIFTR][SHIFTO][SHIFTG][SHIFTR][SHIFTA]
[SHIFTM][SHIFTA][SHIFTS][SPC][SHIFTC]
[SHIFTO][SHIFTN][SHIFTT][SHIFTE][SHIFTN]
[SHIFTI][SHIFTD][SHIFTO][SHIFTS][SPC]
[SHIFTE][SHIFTN][SPC][SHIFTL][SHIFTA]
[SPC][SHIFTC][SHIFTI][SHIFTN][SHIFTT]
[SHIFTA]:"
530 GOSUB5000
540 PRINT:PRINT:PRINT"[SHIFTP]ARA[SPC]
VOLVER[SPC]AL[SPC][SHIFTM][SHIFTE][SHIFTN]
[SHIFTU][SPC][SHIFTP][SHIFTR][SHIFTI]
[SHIFTN][SHIFTC][SHIFTI][SHIFTP][SHIFTA]
[SHIFTL][SPC]PULSAR[SPC]'[SHIFTM]'"
550 GETS$:IFS$=""THEN550
560 IFS$<>"M"THEN550
570 IFS$="M"THEN30
999 REM ************************

1000 REM *   MENU SECUNDARIO (B)   *

1001 REM ************************

1010 PRINT"[CLR]"
1020 GOSUB 2000
1030 PRINT:PRINT:PRINT"[2CRSRR][SHIFTC]
[SHIFTO][SHIFTN][SHIFTT][SHIFTA][SHIFTD]
[SHIFTO][SHIFTR][SHIFTE][SHIFTS][SPC]
[SHIFTD][SHIFTE][SPC][SHIFTL][SHIFTO]
[SHIFTS][SPC][SHIFTP][SHIFTR][SHIFTO]
[SHIFTG][SHIFTR][SHIFTA][SHIFTM][SHIFTA]
[SHIFTS]:"
1040 GOSUB8000
1050 PRINT:PRINT:PRINT"[SHIFTP]ARA[SPC]
VOLVER[SPC]AL[SPC][SHIFTM][SHIFTE][SHIFTN]
[SHIFTU][SPC][SHIFTP][SHIFTR][SHIFTI]
[SHIFTN][SHIFTC][SHIFTI][SHIFTP][SHIFTA]
[SHIFTL][SPC]PULSAR[SPC]'[SHIFTM]'"
1060 GETS$:IFS$=""THEN1060
1070 IFS$<>"M"THEN1060
1080 IFS$="M"THEN30
1499 REM ************************

1500 REM *   MENU SECUNDARIO (C)   *
```

```
1501 REM ****************************

1510 PRINT"[CLR]"CHR$(14)
1520 GOSUB2000
1530 PRINT:PRINT:PRINT"[2CRSRR][SHIFTC]
[SHIFTA][SHIFTR][SHIFTG][SHIFTA][SHIFTR]
[SPC][SHIFTP][SHIFTR][SHIFTO][SHIFTG]
[SHIFTR][SHIFTA][SHIFTM][SHIFTA][SPC]
[SHIFTE][SHIFTL][SHIFTE][SHIFTG][SHIFTI]
[SHIFTD][SHIFTO]:"
1540 PRINT:PRINT:PRINT:PRINT"[4CRSRR]
[SHIFTP]ONGA[SPC]EL[SPC]NUMERO[SPC]D
EL[SPC]PROGRAMA[2SPC]QUE"
1550 PRINT"[4CRSRR]QUIERE[SPC]CARGAR
[2SPC]EN[SPC]EL[SPC]ORDENADOR,[SPC]O
"
1560 PRINT"[4CRSRR]SI[SPC]DESEA[SPC]
VOLVER[SPC]AL[SPC][SHIFTM][SHIFTE][SHIFTN]
[SHIFTU][SPC][SHIFTP][SHIFTR][SHIFTI]
[SHIFTN][SHIFTC][SHIFTI][SHIFTP][SHIFTA]
[SHIFTL]":PRINT"[4CRSRR]PULSE[SPC]´[SHIFTM]
´."
1570 GETS$:IFS$=""THEN1570
1575 PRINTCHR$(142)
1580 GOSUB 10000
1590 IFS$="M"THEN30
1600 PRINTCHR$(14)
1610 GOTO1570
1799 REM ****************************

1800 REM *  MENU SECUNDARIO (D)  *

1801 REM ****************************

1810 PRINT"[CLR]"
1820 GOSUB 2000
1830 PRINT:PRINT:PRINT"[2CRSRR][SHIFTN]
[SHIFTO][SHIFTT][SHIFTA][SHIFTS]:"
1840 GOSUB10600
1850 PRINT:PRINT:PRINT"[SHIFTP]ARA[SPC]
VOLVER[SPC]AL[SPC][SHIFTM][SHIFTE][SHIFTN]
[SHIFTU][SPC][SHIFTP][SHIFTR][SHIFTI]
[SHIFTN][SHIFTC][SHIFTI][SHIFTP][SHIFTA]
[SHIFTL][SPC]PULSAR[SPC]´[SHIFTM]´"
1860 GETS$:IFS$=""THEN1860
1870 IFS$<>"M"THEN1860
1890 IFS$="M"THEN30
1999 REM **** TITULO / CABECERA ****
****
2000 PRINT"[CRSRU][RVSON][40SPC]"
2010 PRINT"[CRSRU][RVSON][9SPC][SHIFTM]
[SHIFTE][SHIFTN][SHIFTU][2SPC][SHIFTC]
[SHIFTA][2SHIFTS][SHIFTE][2SHIFTT][SHIFTE]
[2SPC][SHIFTN][SHIFTO].[SPC]1[10SPC]
"
2020 PRINT"[CRSRU][RVSON][40SPC]"
2030 PRINT"[CRSRU][RVSON][9SPC][SHIFTT]
[SHIFTI][SHIFTT][SHIFTU][SHIFTL][SHIFTO]
:[2SPC][SHIFTV][SHIFTI][SHIFTD][SHIFTE]
[SHIFTO][SPC][SHIFTC][SHIFTA][SHIFTS]
[SHIFTI][SHIFTN][SHIFTO][10SPC]"
2040 PRINT"[CRSRU][RVSON][40SPC]"
2050 RETURN
4999 REM **** PROGRAMAS ************
****
5000 PRINT:PRINT
5010 PRINT"[6CRSRR]1.[SPC][SHIFTT]IR
O[SPC]AL[SPC]BLANCO."
5020 PRINT"[6CRSRR]2.[SPC][SHIFTB]AT
ALLA[SPC]NAVAL."
5030 PRINT"[6CRSRR]3.[SPC][SHIFTM]OS
QUITO."
5040 PRINT"[6CRSRR]4.[SPC][SHIFTB]OM
BARDERO[SPC]LOCO."
5050 PRINT"[6CRSRR]5.[SPC][SHIFTE]L[SPC]
POZO."
5060 PRINT"[6CRSRR]6.[SPC][SHIFTO]BR
AS[SPC]MAESTRAS."
6000 RETURN
```

```
7999 REM **** CONTADORES **********
****
8000 PRINT:PRINT
8010 PRINT"[6CRSRR]1.[SPC]031[SPC]-[SPC]
000"
8020 PRINT"[6CRSRR]2.[SPC]000[SPC]-[SPC]
000"
8030 PRINT"[6CRSRR]3.[SPC]000[SPC]-[SPC]
000"
8040 PRINT"[6CRSRR]4.[SPC]000[SPC]-[SPC]
000"
8050 PRINT"[6CRSRR]5.[SPC]000[SPC]-[SPC]
000"
8060 PRINT"[6CRSRR]6.[SPC]000[SPC]-[SPC]
000"
9000 RETURN
9999 REM **** LOAD´S **************
****
10000 IFS$="1"THENPRINT"[CLR]":LOAD
"TIRO[SPC]AL[SPC]BLANCO"
10010 IFS$="2"THENPRINT"[CLR]":LOAD
"BATALLA[SPC]NAVAL"
10020 IFS$="3"THENPRINT"[CLR]":LOAD
"MOSQUITO"
10030 IFS$="4"THENPRINT"[CLR]":LOAD
"BOMBARDERO[SPC]LOCO"
10040 IFS$="5"THENPRINT"[CLR]":LOAD
"EL[SPC]POZO"
10050 IFS$="6"THENPRINT"[CLR]":LOAD
"OBRAS[SPC]MAESTRAS"
10500 RETURN
10599 REM **** NOTAS **************
****
10600 PRINT:PRINT:PRINT
10610 PRINT"[CRSRR][SHIFTL]OS[SPC]PR
OGRAMAS[SPC]CONTENIDOS[SPC]EN[SPC]ES
TA[SPC]CINTA"
10611 PRINT"[CRSRR]SON[SPC]DE[SPC]TI
PO[SPC]RECREATIVO."
10620 PRINT"[CRSRR][SHIFTE]N[SPC]ALG
UNO[SPC]DE[SPC]ESTOS[SPC]PROGRAMAS[SPC]
ES[2SPC]NECE-"
10621 PRINT"[CRSRR]SARIO[SPC]UTILIZA
R[SPC]EL[SPC][SHIFTJ][SHIFTO][SHIFTY]
[SHIFTS][SHIFTT][SHIFTI][SHIFTC][SHIFTK]
."
10630 PRINT"[CRSRR][SHIFTE]STOS[SPC]
PROGRAMAS[2SPC]ESTAN[SPC]COPIADOS[2SPC]
DE[SPC]LA"
10631 PRINT"[CRSRR]REVISTA[2SPC][SHIFTC]
[SHIFTO][2SHIFTM][SHIFTO][SHIFTD][SHIFTO]
[SHIFTR][SHIFTE][SPC][SPC][SHIFTW][SHIFTO]
[SHIFTR][SHIFTL][SHIFTD],[SPC]DEL[2SPC]
PRIMER"
10632 PRINT"[CRSRR]NUMERO[SPC]HASTA[SPC]
EL[SPC]NUMERO[SPC]6,[SPC]EN[SPC]EL[2SPC]
APAR-"
10633 PRINT"[CRSRR]TADO[SPC][SHIFTV]
IDEO[SPC][SHIFTC]ASINO[SPC]DE[SPC]DI
CHA[SPC]REVISTA."
10800 RETURN
```



Canales abiertos

Cuando tenemos varios canales abiertos y no sabemos cuáles son, si escribimos: SYS 65511 todos los canales se cerrarán automáticamente.

K.H.S. RUN EE.UU.

Defender

David Carlos Manzano Gil
Berriotxoa, nº 9-2º izd.
Barakaldo (BIZCAIA)

VIC-20



Al mando de tu nave espacial DEFENDER tienes que conseguir matar el mayor número posible de nuestros ya queridos los marcianitos.
Pero ojo!, no creáis que todo termina ahí, en la pantalla hay dispersadas diversas minas e incluso algunos marcianos que pueden acabar contigo. También puedes morir desintegrado al hacer un disparo al vacío así que ya sabes y afina bien la puntería.
Bueno, ya se me olvidaba; el programa viene dividido en dos partes y entra en un VIC sin ampliación de ninguna clase.
Es el primer programa que hago y reconozco que no está como se dice perfecto (aunque hay quien dice que la perfección no existe) pero qué pasa, tampoco es una NARRIA!!!
Os felicito por la revista de todos conocida la cual considero que es una gran ayuda para todo el poseedor de un VIC, y aquí dejo mis señas por si alguien de la zona quiere cambiar impresiones...

Parte 1.ª

```
9001 DATA 0 , 24, 24, 60, 126, 255,
255, 126, 60
9002 DATA 1 , 24, 60, 60, 60, 60, 60
, 36, 36
9003 DATA 2 , 24, 24, 60, 126, 255,
255, 255, 255
9004 DATA 3 , 146, 102, 146, 102, 15
0, 102, 150, 102
9005 DATA 4 , 60, 126, 219, 255, 255
, 255, 66, 129
9006 DATA 5 , 60,126,219,219,255,255
,173,173
10000 G1=32768:G2=7168:FORI=0TO511
10001 POKEG2+I,PEEK(G1+I)
10002 NEXTI
10003 FORI=1TO 6 :READC:FORJ=0TO7:RE
ADK
10004 POKEG2+(C*8)+J,K:NEXTJ,I:POKE5
2,28:POKE56,28
11000 POKE631,131:POKE198,1
11500 POKE36879,254
20000 PRINT"[CLR]":  PRINT"[CRSRD][6CRSRR]
DEFENDER"
21000 PRINT"[2CRSRD][5CRSRR]CONTROLE
S"
21100 PRINT"[CRSRD][CRSRR]*Z=IZQUIER
DA"
21110 PRINT"[CRSRD][CRSRR]*X=DERECHA
"
21120 PRINT"[CRSRD][CRSRR]F1=DISPARO
"
21130 PRINT"[CRSRD][CRSRR]ESTE[SPC]P
ROGRAMA[SPC]CARGA[3SPC]LOS[SPC]GRAFI
COS"
21140 PRINT"[CRSRD][3CRSRR]27/09/84[SPC]
DAVID(C)"
```

Parte 2.ª

```
5 GOTO700
6 POKE36879,239
7 G=0 :J=7734:E=7768
8 C=8108:O=7856:T=7812
9 D=7910:F=7954:H=7998
10 PRINT"[CLR]"
30 GETR$
31 IFR$="Z"THENG=1
32 IFR$="X"THENG=2
33 IFR$="[F1]"THEN56
36 FORL=1TO2:POKE7680+INT(RND(1)*200
),4:NEXTL
37 FORL=1TO3:POKE7680+INT(RND(1)*200
),32:NEXTL
38 FORY=1TO2:POKE7680+INT(RND(1)*200
),5:NEXTY
39 FORY=1TO6:POKE7680+INT(RND(1)*200
),32:NEXTY
55 GOTO66
56 REM*DISPARO*
57 FORW=1TO22:FORP=1TO22
58 V=C-(22*W):B=C-(22*P)
59 POKEV,1:POKEB,32
60 IFPEEK(VANDB)=0THENGOTO 2000
61 IFPEEK(VANDB)=3THENGOTO2200
62 IFPEEK(VANDB)=5THENGOTO2400
63 IFPEEK(VANDB)=4THENGOTO2600
65 NEXTP,W
66 IFG=1THENC=C-1:POKEC+1,32
70 IFG=2THENC=C+1:POKEC-1,32
80 IFC<8099ANDG=1THENC=8098
90 IFC>8118ANDG=2THENC=8119
110 POKEC,2
115 FORY=1TO 100:NEXTY
120 REM*MARCIANO*
121 U=INT(RND(1)*3)
122 IFU=1THENJ=J-1:POKEJ+1,32
123 IFU=2THENJ=J+1:POKEJ-1,32
124 IFJ<7747THENJ=7746
125 IFJ>7766THENJ=7767
126 POKEJ,0
127 Z=INT(RND(1)*3)
130 IFZ=1THENE=E+1:POKEE-1,32
131 IFZ=2THENE=E-1:POKEE+1,32
132 IFE<7769THENE=7768
133 IFE>7788THENE=7789
134 POKEE,0
135 X=INT(RND(1)*3)
136 IFX=1THENT=T+1:POKET-1,32
137 IFX=2THENT=T-1:POKET+1,32
138 IFT<7813THENT=7812
139 IFT>7822THENT=7821
140 POKET,0
141 Y=INT(RND(1)*3)
142 IFY=1THENO=O-1:POKEO+1,32
143 IFY=2THENO=O+1:POKEO-1,32
144 IFO<7857THENO=7856
145 IFO>7876THENO=7877
146 POKEO,0
```

```
147 W=INT(RND(1)*3)
148 IFW=1THEND=D-1:POKED+1,32
149 IFW=2THEND=D+1:POKED-1,32
150 IFD<7901THEND=7900
151 IFD>7920THEND=7921
152 POKED,3
153 Q=INT(RND(1)*3)
154 IFQ=1THENF=F-1:POKEF+1,32
155 IFQ=2THENF=F+1:POKEF-1,32
156 IFF<7945THENF=7944
157 IFF>7964THENF=7965
158 POKEF,3
159 P=INT(RND(1)*3)
160 IFP=1THENH=H-1:POKEH+1,32
161 IFP=2THENH=H+1:POKEH-1,32
162 IFH<7989THENH=7988
163 IFH>8008THENH=8009
164 POKEH,3
670 PRINT"[HOM][RVSON][WHT][21CRSRD]
PUNTOS";PU
690 GOTO30
700 POKE36869,255:PRINT"[CLR]":POKE3
6879,171
710 PRINT"[CRSRD][7CRSRR][RVSON]DEFE
NDER"
720 PRINT"[5CRSRR][RVSON][12COMMU]"
730 PRINT"[2CRSRD][5CRSRR][RVSON]INS
TRUCCIONES"
740 PRINT"[3CRSRD][CRSRR][RVSON]*Z=I
ZQUIERDA"
750 PRINT"[CRSRR][RVSON]*X=DERECHA"
760 PRINT"[CRSRR][RVSON]*F1=DISPARA
762 PRINT"[CRSRD][CRSRR][RVSON]ATENC
ION[SPC]LAS[SPC]MINAS[5SPC]Y[SPC]ALG
UNOS[SPC]MARCIANOS[2SPC]TE[SPC]DESTR
UYEN!"
770 PRINT"[4CRSRD][5CRSRR][RVSON]BUE
NA[SPC]SUERTE"
775 FORU=1TO2900:NEXTU
780 GOTO6
2000 POKE36877,220:FORY=15TO0STEP-1
2010 POKE36878,Y:FORU=1TO500
2015 NEXTU,Y
2020 POKE36877,0:POKE36878,0
2100 PRINT"[HOM][7CRSRR][7CRSRD][RVSON]
[WHT]GAME[SPC]OVER":FORR=1TO3000:NEX
TR:RUN
2110 GOTO6
2200 POKE36878,15:POKE36875,232:POKE
36876,230
2210 FORT=1TO150:NEXTT
2220 POKE36876,0:POKE36875,0:POKE368
78,0
2230 PU=PU+10
2250 GOTO6
2400 POKE36878,15:POKE36874,201
2410 POKE36875,201:POKE36876,201
2420 FORT=1TO150:NEXTT
2430 POKE36876,0:POKE36875,0
2440 POKE36874,0:POKE36878,0
2450 PU=PU+30
2460 GOTO6
2600 POKE36878,15:POKE36874,241:POKE
36875,241
2610 POKE36876,241
2620 FORT=1TO150:NEXTT
2630 POKE36876,0:POKE36875,0
2640 POKE36874,0:POKE36878,0
2650 PU=PU+50
4000 GOTO6
```

La ruleta

Sergi Arbós Mauri
C/ Forn del Vidre 16-3º-4ª
Vilanova i La Geltru (Barcelona)

C-64

Este programa funciona igual que una ruleta normal, sólo que no se puede apostar ni al 0 ni al doble 00; tampoco se puede apostar a dos cosas a la vez (Ej.: impar y rojo).

El programa funciona así:

Primero sale la presentación, luego te pregunta cuánto deseas apostar, luego a qué y al final sale el número extraído por la ruleta, y si has ganado te dice cuánto y suena una musiquilla; si no, suena otra musiquilla anunciándote que has perdido. Al final te pregunta si quieres jugar otra vez.

1-20 Presentación en pantalla
30 Pregunta cuánto quieres apostar
50-360 Pregunta a qué quieres apostar y lo guarda en variables
360-555 Se elige un número al azar
600-830 Muestra el número elegido
1000-1030 Comprueba si has ganado
1040 Si has perdido te lo anuncia
1045 Te manda a la subrutina de sonido
1100 Si has ganado te lo anuncia
1101-1104 Te informan de lo que has ganado
1105 Te manda a la otra subrutina de sonido
1110 Te pregunta si quieres volver a jugar
1130-1404 Se despide
2000-2300 Primera subrutina de sonido
2900-4500 Segunda subrutina de sonido

```
1 REM
2 POKE53280,2
3 POKE53281,6
4 PRINT"[CLR]"
5 PRINT"[GRN]"
6 PRINT"[4SPC]LA[SPC]RULETA[SPC]AMER
ICANA"
7 PRINT
8 PRINT"[3SPC]POR[SPC]SERGI[SPC]ARBO
S[SPC]MAURI"
9 PRINT
10 PRINT"[WHT]"
11 PRINT"[RVSON][SHIFTQ][SHIFTW][SHIFTQ]
[SHIFTW][SHIFTQ][SHIFTW][SHIFTQ][SHIFTW]
[SHIFTQ][SHIFTW][SHIFTQ][SHIFTW][SHIFTQ]
[SHIFTW][SHIFTQ][SHIFTW][SHIFTQ][SHIFTW]
```

```
[SHIFTQ][SHIFTW][SHIFTQ][SHIFTW][SHIFTQ]
[SHIFTW][SHIFTQ][SHIFTW][SHIFTQ][SHIFTW]
[SHIFTQ][SHIFTW][SHIFTQ][SHIFTW]"
12 PRINT"[WHT]"
13 PRINT"[RVSON][SHIFTQ][SHIFTW][SHIFTQ]
[SHIFTW][SHIFTQ][SHIFTW][SHIFTQ][SHIFTW]
[SHIFTQ][SHIFTW][SHIFTQ][SHIFTW][SHIFTQ]
[SHIFTW][SHIFTQ][SHIFTW][SHIFTQ][SHIFTW]
[SHIFTQ][SHIFTW][SHIFTQ][SHIFTW][SHIFTQ]
[SHIFTW][SHIFTQ][SHIFTW][SHIFTQ][SHIFTW]
[SHIFTQ][SHIFTW][SHIFTQ][SHIFTW]"
14 PRINT"[SHIFTA][SHIFTS][SHIFTZ][SHIFTX]
[SHIFTA][SHIFTS][SHIFTZ][SHIFTX][SHIFTA]
[SHIFTS][SHIFTZ][SHIFTX][SHIFTA][SHIFTS]
[SHIFTZ][SHIFTX][SHIFTA][SHIFTS][SHIFTZ]
[SHIFTX][SHIFTA][SHIFTS][SHIFTZ][SHIFTX]
[SHIFTA][SHIFTS][SHIFTZ][SHIFTX][SHIFTA]
[SHIFTS][SHIFTZ][SHIFTX]"
15 PRINT"[RVSON][SHIFTA][SHIFTS][SHIFTZ]
[SHIFTX][SHIFTA][SHIFTS][SHIFTZ][SHIFTX]
[SHIFTA][SHIFTS][SHIFTZ][SHIFTX][SHIFTA]
[SHIFTS][SHIFTZ][SHIFTX][SHIFTA][SHIFTS]
[SHIFTZ][SHIFTX][SHIFTA][SHIFTS][SHIFTZ]
[SHIFTX][SHIFTA][SHIFTS][SHIFTZ][SHIFTX]
[SHIFTA][SHIFTS][SHIFTZ][SHIFTX]"
16 PRINT
17 PRINT"LE[SPC]DESEO[SPC]MUCHA[SPC]
SUERTE"
18 FORI=1TO1500
19 NEXTI
20 PRINT"[CLR]"
30 INPUT"CUANTO[SPC]DESEAS[SPC]APOST
AR";H1
35 PRINT"[CLR]"
45 PRINT
50 PRINT"QUIERES[SPC]APOSTAR[SPC]A[SPC]
UN[SPC]NUMERO(N)?"
60 PRINT"A[SPC]UN[SPC]COLOR[SPC](C)?
"
70 PRINT"A[SPC]PAR[SPC]O[SPC]IMPAR(I
)?"
80 PRINT"A[SPC]PASSE[SPC]O[SPC]MANQU
E(R)?"
90 INPUTJ$
95 IFJ$="N"THEN GOTO200
100 IFJ$="C"THEN GOTO250
110 IFJ$="I"THEN GOTO350
120 IFJ$="R"THENGOTO300
123 END
200 PRINT"[CLR]":INPUT"A[SPC]QUE[SPC]
NUMERO[SPC]QUIERES[SPC]APOSTAR";S2
210 GOTO500
250 PRINT"[CLR]":INPUT"A[SPC]QUE[SPC]
COLOR[SPC]QUIERES[SPC]APOSTAR,AL[SPC]
ROJO(A1),O[SPC]AL[SPC]NEGRO(E)";P$
260 GOTO500
300 PRINT"[CLR]":INPUT"QUIERES[SPC]A
POSTAR[SPC]A[SPC]PASSE(J)O[SPC]MANQU
E(M)";O$
310 GOTO500
350 PRINT"[CLR]":INPUT"QUIERES[SPC]A
POSTAR[SPC]A[SPC]PAR(U)O[SPC]A[SPC]I
MPAR(B1)";K$
360 GOTO500
500 A=INT(38*RND(1))+1
510 IFA=37THENA=0
520 IFA=38THENA=00
525 IFA=0ORA=00 THEN GOTO800
530 Q=A-2*INT(A/2)
550 IFQ=1THEN GOTO 600
555 IFQ=0THEN GOTO700
600 PRINT"[CLR]":PRINT"EL[SPC]NUMERO
[SPC]EXTRAIDO[SPC]POR[SPC]LA[SPC]RUL
ETA"
601 PRINT
602 PRINT"[15SPC]ES[SPC]EL":PRINT
603 PRINT"[RVSON]"TAB(16)A:PRINT
609 PRINT"[15SPC]IMPAR"
615 PRINT
620 PRINT"[15SPC]NEGRO"
625 PRINT
630 IFA>19 THEN GOTO650
640 PRINT"[15SPC]MANQUE"
645 G$="M"
647 GOTO660
650 PRINT"[15SPC]PASSE"
651 PRINT
655 G$="J"
660 D$="B1"
670 V$="E"
680 GOTO1000
700 PRINT"[CLR]":PRINT"EL[SPC]NUMERO
[SPC]EXTRAIDO[SPC]POR[SPC]LA[SPC]RUL
ETA":PRINT
701 PRINT"[14SPC]ES[SPC]EL":PRINT
702 PRINT"[RVSON]"TAB(16)A:PRINT
709 PRINT
710 PRINT"[14SPC]PAR"
715 PRINT
720 PRINT"[14SPC]ROJO"
725 PRINT
730 IFA>19THEN GOTO750
740 PRINT"[14SPC]MANQUE"
745 G$="M"
746 GOTO760
750 PRINT"[14SPC]PASSE"
751 PRINT
755 G$="J"
760 D$="U"
770 V$="A1"
780 GOTO1000
800 PRINT"EL[SPC]NUMERO[SPC]EXTRAIDO
[SPC]POR[SPC]LA[SPC]RULETA[SPC]":PRI
NT
810 PRINT"[13SPC]ES[SPC]EL":PRINT
820 PRINT"[RVSON]"TAB(16)A:PRINT
825 PRINT"[13SPC]VERDE"
826 PRINT
830 GOTO1040
RUN
1000 IFD$=K$THEN GOTO1100
1010 IFV$=P$ THENGOTO1100
1020 IFG$=O$THEN GOTO1100
1030 IFA=K2THEN GOTO1100
1040 PRINT"LO[SPC]SIENTO[SPC]HAS[SPC]
PERDIDO"
1045 GOSUB2900
1050 GOTO1110
1100 PRINT:PRINTTAB(3)"ENHORABUENA,[SPC]
HAS[SPC]GANADO"
1101 IFJ$="N"THEN H2=H1*37
1102 IFJ$="C"ORJ$="I"ORJ$="R" THEN H
2=H1*2
1103 PRINT
1104 PRINTTAB(5)H2"PESETAS!"
1105 GOSUB2000
1110 PRINT:INPUT"OTRA(S/N)";F$
1120 IFF$="S"THEN GOTO 1
1130 PRINT"[CLR]"
1400 FORI=1TO50
1401 PRINT"VUELVA[SPC]AL[SPC]GRAN[SPC]
CASINO"
1402 NEXTI
1403 PRINT"[CLR]"
1404 END
2000 FORL=54272TO54296:POKEL,0:NEXT
2010 V1=54296:W1=54276:A3=54277:S1=5
4278:H3=54273:L=54272
2020 POKEV1,15:POKEA3,190:POKES,89
2030 POKEH3,43:POKEL,52
2040 POKEW1,33:FORT=1TO500:NEXT
2050 POKEW1,32
2060 POKEH3,51:POKEL,97
2070 POKEW1,33:FORT=1TO500:NEXT
2080 POKEW1,32
2090 POKEH3,57:POKEL,172
2100 POKEW1,33:FORT=1TO500:NEXT
2110 POKEW1,32
```

```
2120 POKEH3,43:POKEL,52
2130 POKEW1,33:FORT=1TO500:NEXT
2140 POKEW1,32
2150 POKEH3,51:POKEL,97
2160 POKEW1,33:FORT=1TO1000:NEXT
2170 POKEW1,32
2180 POKEH3,57:POKEL,172
2190 POKEW1,33:FORT=1TO800:NEXT
2200 POKEW1,32
2300 RETURN
2900 FORL=54272TO54296:POKEL,0:NEXT
3000 V1=54296:W1=54276:A3=54277:S=54
278:H3=54273:L=54272
3010 POKEV1,15:POKEA3,190:POKES,89
3050 POKEH3,36:POKEL,85
3060 POKEW1,33:FORT=1TO500:NEXT
3070 POKEW1,32
3080 POKEH3,38:POKEL,126
3090 POKEW1,33:FORT=1TO500:NEXT
3100 POKEW1,32
3110 POKEH3,43:POKEL,52
3120 POKEW1,33:FORT=1TO1000:NEXT
3130 POKEW1,32
4020 POKEH3,34:POKEL,75
4030 POKEW1,33:FORT=1TO500:NEXT
4040 POKEW1,32
4500 RETURN
```

Base lunar

Luis García Truebano
Rosal, 48-4º B
83009 OVIEDO

VIC-20

Os envío una nueva versión de "BASE LUNAR", original y distinta a la ya aparecida para C-64. El programa, en su listado, no contiene instrucciones REM ya que ocupa la práctica totalidad de la memoria del VIC-20 sin ampliar. El juego os lo envío yo como suscriptor de la revista, aunque su realización corre a cargo de mi hermano Jorge Juan, que tiene 12 años. Ahí va la explicación del juego:

0-40: Instrucciones
100-200: Caracteres programables
203-204: Variables
205-310: Dibujo del paisaje lunar
320-375: Movimiento base lunar
380-450: Movimiento de la nave (bajada)
3000-3030: Destrucción de una de tus naves
3040-3120: Fin de partida
3500-3520: Movimiento del astronauta
3550-3590: Movimiento de los ovnis
3595-3660: Movimiento de tu nave (subida)
4000-4170: Datas musicales
4500-4510: Indica el número de pantalla

Para abandonar la nave nodriza debes de pulsar "A"

Y algo importante: una vez ejecutado el programa, la escasez de memoria sólo permite la grabación del mismo en una cinta virgen con la instrucción SAVE a secas, sin poner nombre al programa, ya que de hacerlo nos encontraríamos con el mensaje "OUT OF MEMORY". Para cargar el programa en otra cinta con su nombre, por ejemplo "SAVE BASE LUNAR", debemos efectuar esta operación ANTES de ejecutar el programa.

```
10 POKE36879,8:PRINT"[CLR][2SPC][RVSON]
[YEL]*BASE[SPC]LUNAR*[RVSOFF]
20 PRINT"[WHT][2CRSRD]GUIA[SPC]TU[SPC]
NAVE[SPC]CON:":PRINT"[CRSRD]´:´IZQ.[SPC]
Y[SPC]´;´DER.
35 PRINT"[2SPC][RVSON][3CRSRD]PULSA[SPC]
UNA[SPC]TECLA[RVSOFF]
40 GETA$:IFA$=""THEN40
100 PRINT"[CLR]"
110 POKE52,28:POKE56,28:CLR
120 FORX=7168TO7679:POKEX,PEEK(X+256
00):NEXTX
130 POKE36869,255
140 FORL=7168TO7270:READD:POKEL,D:NE
XTL
150 DATA224,224,248,172,252,248,248,
0
160 DATA255,60,126,90,60,231,255,0
170 DATA3,3,31,53,63,31,31,0
180 DATA255,0,0,0,0,0,255,0
191 DATA24,60,126,90,126,60,24,231
192 DATA255,255,255,255,255,255,255,
255
193 DATA28,60,124,126,62,28,24,0
194 DATA129,195,255,255,60,60,60,126

195 DATA136,204,238,255,255,255,255,
255
196 DATA126,102,102,231,231,231,231,
195
197 DATA0,56,124,146,254,84,0,0
198 DATA0,24,8,56,8,20,20,0
200 DATA146,84,56,254,56,84,146,0
203 MA=3:SC=0:MO=32:V=36878:S3=36876
:LK=1:POKE788,194
204 PRINT"[CLR]":MI=0:GOSUB4500
205 FORSA=8120TO8141
210 POKESA,8:NEXT
220 FORSA=8142TO8185
230 POKESA,5:NEXT
240 FORSA=1TO75
250 X=INT(0+310*RND(1))
260 POKE38444+X,6:POKE7724+X,6:NEXT
270 W=INT(0+21*RND(1))
280 POKE8076+22+W,7
290 W=INT(0+21*RND(1))
293 IFPEEK(8076-22+W)=7THENW=W+1
295 POKE8076+W,7:POKE8076+W+22,9
300 FORA=38400TO38421
310 POKEA,4:NEXTA
320 FORN=7683TO7701
350 POKEN,0:POKEN-1,1:POKEN-2,2:POKE
N-3,32
360 GETA$:IFA$="A"THEN380
370 FORT=1TO150:NEXTT:NEXTN
375 POKE7680+21,32:POKE7680+20,32:PO
KE7680+19,32:GOTO320
380 POKEN-1,3:AD=N+21:POKEAD,4:FORT=
1TO500:NEXT
390 GETB$
410 IFB$=":"THENB=AD:AD=AD+21:GOTO43
5
```

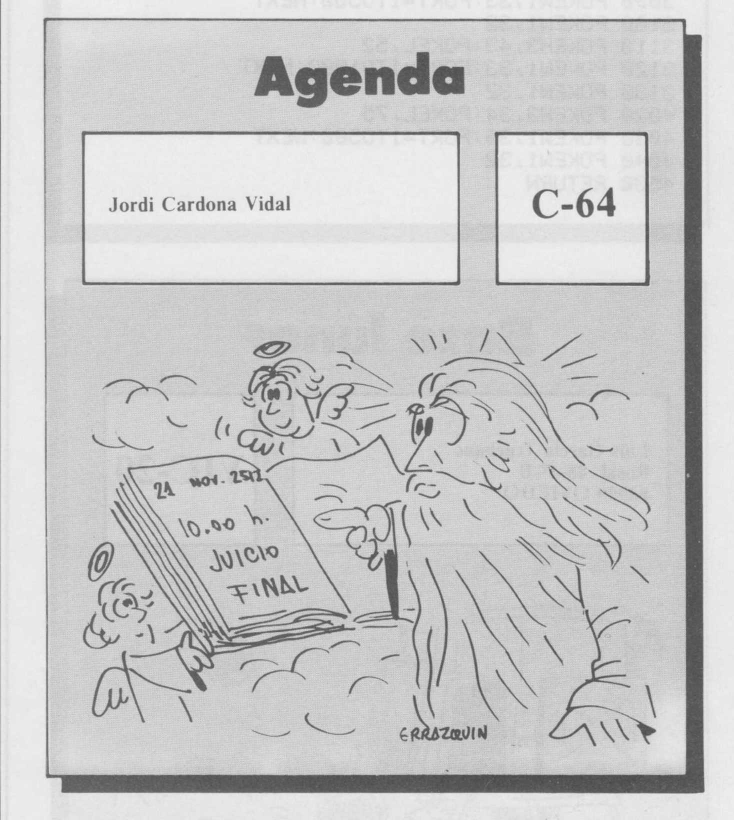
```
420 IFB$=";"THENB=AD:AD=AD+23:GOTO43
5
425 B=AD:AD=AD+22
435 IFPEEK(AD)=7THENAB=B:GOTO3500
437 POKEV,15:POKES3,223:FORT=1TO100:
NEXT:POKES3,0:FORSA=1TO5
438 F=INT(0+278*RND(0)):IFF=ADORPEEK
(7724+F)=6THEN438
439 POKE38444+F,6:POKE7724+F,6:NEXT
440 IFPEEK(AD)=6ORPEEK(AD)=8THEN3000
450 POKEAD,4:POKEB,32:FORT=1TO400:NE
XTT:GOTO390
3000 POKEB,32:POKEAD,12
3005 FORRT=15TO0STEP-1:POKEV,RT:POKE
36877,207
3010 FORSA=1TO50:NEXT:NEXT:POKE36877
,0
3020 MA=MA-1:IFMA<1THEN3040
3030 GOTO204
3040 PRINT"[CLR]":POKE36869,240:PRIN
T"[YEL][4CRSRD][7CRSRR][RVSON]GAME[SPC]
OVER[RVSOFF]"
3050 PRINT"[2CRSRD]HAS[SPC]CONSEGUID
O:[RVSON]"SC"[RVSOFF][HOM][10CRSRD]P
UNTOS
3070 GOSUB4000
3080 PRINT"[CRSRD][WHT][3SPC]OTRA[SPC]
VEZ?(S/N)
3090 GETC$:IFC$=""THEN3090
3100 IFC$="S"THENRESTORE:GOTO100
3110 IFC$<>"N"THEN3090
3120 PRINT"[CRSRD].........ADIOS....
....":END
3500 SC=SC+150
3505 POKE8119-MI,11:FORRT=1TO50:NEXT
:POKE8119-MI,MO:MI=MI+1:MO=32
3510 IFPEEK(8119-MI-22)=4ORPEEK(8119
-MI-44)=4THEN3530
3515 IFPEEK(8119-MI)=9THENMO=9
3517 IFPEEK(8119-MI)=7THENMO=7
3518 POKEV,15:POKES3,175:FORT=1TO50:
NEXT:POKES3,0
3520 GOTO3505
3530 FORSA=7702TO8053:POKESA,32:NEXT

3540 FORSA=38444TO38774:POKESA,1:NEX
T
3550 FORSA=7790TO7808STEP4:POKESA+Y,
10:POKESA-1+Y,32:NEXT:Y=Y+1:IFY>3THE
NY=0
3570 FORSA=7857TO7876STEP4:POKESA+Y,
10:POKESA+Y-1,32:NEXT
3580 FORSA=7922TO7941STEP4:POKESA+Y,
10:POKESA-1+Y,32:NEXT
3590 FORSA=7989TO8007STEP4:POKESA+Y,
10:POKESA-1+Y,32:NEXT
3595 IFPEEK(AB)=10THENPOKENA,32:POKE
AB,12:GOTO3005
3600 GETD$:IFD$=":"THENNA=AB:AB=AB-2
3:GOTO3630
3610 IFD$=";"THENNA=AB:AB=AB-21:GOTO
3630
3620 NA=AB:AB=AB-22
3630 IFPEEK(AB)=10ORPEEK(AB)=0ORPEEK
(AB)=1THENPOKENA,32:POKEAB,12:GOTO30
05
3640 IFPEEK(AB)=3THENPOKENA,32:POKEA
B,1:FORT=1TO500:NEXT:SC=SC+200:LK=LK
+1:GOTO204
3650 POKES3,223:FORT=1TO100:NEXT:POK
ES3,0
3660 POKENA,32:POKEAB,4:FORT=1TO500:
NEXT:GOTO3550
4000 POKEV,15
4010 READI
4020 IFI=-1THENPOKES3,0:RETURN
4030 READK
4040 POKES3,I
4050 FORT=1TOK:NEXT
4060 POKES3,0
4070 FORT=1TO20:NEXT
4080 GOTO4010
4100 DATA0,225,360,225,360,225,240
4110 DATA228,120,231,360,231,240
4120 DATA228,120,231,240,232,120
4130 DATA235,720,240,360,235,360
4140 DATA231,360,225,360,235,240
4150 DATA232,120,231,240,228,120
4160 DATA225,480
4170 DATA-1
4500 FORT=1TO3:PRINT"[9CRSRD][4CRSRR]
[WHT]P[RVSON]AN[RVSOFF]T[RVSON]ALLA"
;LK:FORSA=1TO100:NEXTSA
4510 FORSA=1TO50:PRINT"[CLR]":NEXT:N
EXTT:RETURN
```

Agenda

Jordi Cardona Vidal

C-64



Este programa tarda en cargarse en cassette 3 minutos 15 segundos. Dispone de un fichero secuencial con la información de los teléfonos grabados anteriormente, al ejecutar el programa por primera vez dicho fichero no está aún abierto. Es aconsejable guardar el fichero en una cinta independiente de la que está grabado el programa, y además se aconseja grabar el fichero (una vez abierto) al principio del programa, hacer todas las modificaciones necesarias y guardar de nuevo el fichero modificado al final de la ejecución, recuerda rebobinar la cinta donde tienes el fichero. Lógicamente si no has hecho modificaciones, sólo has consultado, no es necesario guardarlo de nuevo.

```
9 POKE53281,1
10 POKE53280,1
20 GOSUB13000
22 DIMNOM$(301),COG$(301),TEL$(301),
B$(20)
24 POKE650,255
30 PRINT"[CLR][2CRSRD][9CRSRR][RVSON]
[2SPC]AGENDA[SPC]TELEFONICA[2SPC]"
40 PRINT"[CRSRD][12CRSRR][RVSON][SPC]
MENU[SPC]GENERAL[SPC]"
```

```
50 PRINT"[2CRSRD][6CRSRR]1[SPC]-[SPC]
INSTRUCCIONES
55 PRINT"[CRSRD][6CRSRR]2[SPC]-[SPC]
LEER[SPC]EL[SPC]FICHERO"
60 PRINT"[CRSRD][6CRSRR]3[SPC]-[SPC]
ENTRAR[SPC]UN[SPC]TELEFONO"
70 PRINT"[CRSRD][6CRSRR]4[SPC]-[SPC]
BORRAR[SPC]UN[SPC]TELEFONO"
80 PRINT"[CRSRD][6CRSRR]5[SPC]-[SPC]
BUSCAR[SPC]UN[SPC]TELEFONO"
85 PRINT"[CRSRD][6CRSRR]6[SPC]-[SPC]
GUARDAR[SPC]EL[SPC]FICHERO"
88 PRINT"[CRSRD][6CRSRR]7[SPC]-[SPC]
FINALIZAR"
90 PRINT"[2CRSRD][6CRSRR][RVSON][SPC]
PULSA[SPC]LA[SPC]OPCION[SPC]QUE[SPC]
DESEES[SPC]"
100 GETA$
110 IFA$="1"THEN GOSUB300:GOTO30
115 IFA$="2"THEN GOSUB6000:GOTO30
120 IFA$="3"THEN GOSUB700:GOTO30
130 IFA$="4"THEN GOSUB4000:GOTO30
140 IFA$="5"THEN GOSUB3000:GOTO30
150 IFA$="6"THEN GOSUB5000:GOTO30
160 IFA$="7"THENPRINT"[CLR]":END
170 GOTO100
300 PRINT"[CLR][2CRSRD][9CRSRR][RVSON]
[2SPC]AGENDA[SPC]TELEFONICA[2SPC]"
305 PRINT"[CRSRD][12CRSRR][RVSON][SPC]
INSTRUCCIONES[SPC]"
310 PRINT"[CRSRD][3SPC]LA[SPC]CAPACI
DAD[SPC]DE[SPC]ESTA[SPC]AGENDA[SPC]E
S[SPC]DE[SPC]300"
320 PRINT"[CRSRU]NOMBRES,[SPC]DISPOS
ES[SPC]DE[SPC]SEIS[SPC]OPCIONES:"
321 PRINT"[CRSRD][8CRSRR]VER[SPC]LAS
[SPC]INSTRUCCIONES"
322 PRINT"[8CRSRR]LEER[SPC]EL[SPC]FI
CHERO"
330 PRINT"[8CRSRR]ENTRAR[SPC]UN[SPC]
TELEFONO"
340 PRINT"[8CRSRR]BORRAR[SPC]UN[SPC]
TELEFONO"
350 PRINT"[8CRSRR]BUSCAR[SPC]UN[SPC]
TELEFONO"
352 PRINT"[8CRSRR]GUARDAR[SPC]EL[SPC]
FICHERO"
360 PRINT"[CRSRD][3SPC]LOS[SPC]TELEF
ONOS[2SPC]ESTANT[SPC]GUARDADOS[SPC]E
N[SPC]UN"
365 PRINT"[CRSRU]UNICO[SPC]FICHERO[SPC]
SECUENCIAL.[SPC]LA[SPC]PRIMERA[SPC]V
EZ"
370 PRINT"[CRSRU]QUE[SPC]ES[SPC]UTIL
ITZADO[SPC]EL[SPC]PROGRAMA[SPC]EL[SPC]
FICHERO"
375 PRINT"[CRSRU]NO[SPC]ESTA[SPC]AUN
[SPC]CREADO,SI[SPC]PIDES[SPC]LA[SPC]
OPCION[SPC]DE"
380 PRINT"[CRSRU]LEER[SPC]EL[SPC]FIC
HERO[SPC]DARA[SPC]ERROR."
385 PRINT"[2CRSRD][4CRSRR][RVSON][SPC]
PULSA[SPC]UNA[SPC]TECLA[SPC]PARA[SPC]
CONTINUAR[SPC]"
390 GETA$:IFA$=""THEN390
395 PRINT"[CLR][2CRSRD][9CRSRR][RVSON]
[2SPC]AGENDA[SPC]TELEFONICA[2SPC]"
400 PRINT"[CRSRD][12CRSRR][RVSON][SPC]
INSTRUCCIONES[SPC]"
402 PRINT"[CRSRD][3SPC]CUANDO[SPC]EN
TRES[SPC]UN[SPC]TELEFONO[SPC]DEBERAS
[SPC]DAR"
404 PRINT"[CRSRU]TAMBIEN[SPC]EL[SPC]
NOMBRE[SPC]Y[SPC]APELLIDOS[2SPC]DE[SPC]
LA[SPC]PER-"
406 PRINT"[CRSRU]SONA[SPC]A[SPC]LA[SPC]
QUE[SPC]PERTENECE.CUANDO[SPC]HAYAS[SPC]
EN-"
408 PRINT"[CRSRU]TRADO[SPC]LOS[SPC]D
ATOS,SI[SPC]SON[SPC]CORRECTOS[SPC]AP
RIETA"
410 PRINT"[CRSRU]RETURN,SINO[SPC]LA[SPC]
TECLA[SPC](I),LA[SPC]QUAL[SPC]TE[SPC]
PER-"
412 PRINT"[CRSRU]MITIRA[SPC]MODIFICA
R[SPC]EL[SPC]ERROR."
414 PRINT"[CRSRD][2SPC]SI[SPC]LOS[SPC]
DATOS[2SPC]SON[SPC]CORRECTOS[SPC]PUE
DES[SPC]EN-"
416 PRINT"[CRSRU]TRAR[SPC]OTRO[SPC]N
OMBRE[SPC]APRETANDO[SPC]RETURN,[SPC]
GUAR-"
418 PRINT"[CRSRU]DANDO[SPC]ESTE[SPC]
EN[SPC]UNA[SPC]LISTA[SPC]ORDENADA."
420 PRINT"[CRSRD][3SPC]PARA[SPC]BORR
AR[SPC]UN[SPC]TELEFONO[2SPC]DEBES[SPC]
DE[SPC]EN-"
422 PRINT"[CRSRU]TRAR[SPC]UNA[SPC]RE
FERENCIA[SPC]DEL[SPC]NOMBRE[SPC]DE[SPC]
LA[SPC]PER"
424 PRINT"[CRSRU]SONA[SPC]Y[SPC]AUTO
MATICAMENTE[SPC]QUEDARA[SPC]BORRADO.
"
426 PRINT"[CRSRU]SI[SPC]HAY[SPC]DOS[SPC]
REFERENCIAS[SPC]IGUALES[SPC]SE[SPC]B
ORRA-"
428 PRINT"[CRSRU]RA[SPC]EL[SPC]PRIME
RO[SPC]DE[SPC]LA[SPC]LISTA."
430 PRINT"[2CRSRD][4CRSRR][RVSON][SPC]
PULSA[SPC]UNA[SPC]TECLA[SPC]PARA[SPC]
CONTINUAR[SPC]"
432 GETA$:IFA$=""THEN432
440 PRINT"[CLR][2CRSRD][9CRSRR][RVSON]
[2SPC]AGENDA[SPC]TELEFONICA[2SPC]"
450 PRINT"[CRSRD][12CRSRR][RVSON][SPC]
INSTRUCCIONES[SPC]"
460 PRINT"[CRSRD][3SPC]PARA[SPC]BUSC
AR[SPC]EL[SPC]TELEFONO[SPC]DE[SPC]UN
A[SPC]PERSO-"
470 PRINT"[CRSRU]NA[SPC]TENDRAS[SPC]
QUE[SPC]ENTRAR[SPC]UNA[SPC]REFERENCI
A[SPC]TAL"
480 PRINT"[CRSRU]COMO[SPC]LA[SPC]PRI
MERA[SPC]O[SPC]LAS[SPC]PRIMERAS[SPC]
LETRAS[SPC]DE"
490 PRINT"[CRSRU]SU[SPC]APELLIDO.ENT
ONCES[SPC]VERAS[SPC]TODOS[SPC]LOS[SPC]
TE-"
500 PRINT"[CRSRU]LEFONOS[SPC]DE[SPC]
LAS[SPC]PERSONAS[SPC]CON[SPC]IDENTIC
A[SPC]RE-"
510 PRINT"[CRSRU]FERENCIA."
520 PRINT"[CRSRD][3SPC]SI[SPC]QUIERE
S[SPC]VER[SPC]TODOS[SPC]LOS[SPC]TELE
FONOS[SPC]DE"
530 PRINT"[CRSRU]LA[SPC]LISTA[SPC]AP
RIETA[SPC]RETURN[SPC]Y[SPC]LOS[SPC]V
ERAS[SPC]ALFA-"
540 PRINT"[CRSRU]BETICMENTE,[SPC]POR
[SPC]PANTALLAS[SPC]DE[SPC]16[SPC]NOM
BRES."
550 PRINT"[3SPC]CUANDO[SPC]HAYAS[2SPC]
ACABADO[SPC]DE[SPC]CONSULTAR[SPC]SI"
560 PRINT"[CRSRU]HAS[SPC]MODIFICADO[SPC]
EL[SPC]FICHERO[SPC]RECUERDA[SPC]REBO
-"
570 PRINT"[CRSRU]BINAR[SPC]LA[SPC]CI
NTA[SPC]HASTA[SPC]EL[SPC]LUGAR[SPC]A
DECUADO[SPC]Y"
580 PRINT"[CRSRU]APRETAR-6-PARA[SPC]
GUARDAR[SPC]EL[SPC]NUEVO[SPC]FICHERO
."
660 PRINT"[CRSRD][4CRSRR][RVSON][SPC]
PULSA[SPC]UNA[SPC]TECLA[SPC]PARA[SPC]
CONTINUAR[SPC]"
670 GETA$:IFA$=""THEN670
680 RETURN
```

```
700 IFY<300THEN714
702 PRINT"[CLR]"
703 OP=18
704 FORWQ=0TO25
705 FORX=1TO250:NEXT
706 PRINTCHR$(OP)"[HOM][12CRSRD][6CRSRR]
[SPC]SOLO[SPC]PUEDO[SPC]ADMITIR[SPC]
300[SPC]NOMSBRES"
707 IFOP=18THENOP=146:GOTO709
708 IFOP=146THENOP=18
709 NEXTWQ
710 RETURN
714 PRINT"[CLR][2CRSRD][9CRSRR][RVSON]
[2SPC]AGENDA[SPC]TELEFONICA[2SPC]"
715 PRINT"[CRSRD][9CRSRR][RVSON][2SPC]
ENTRAR[SPC]UN[SPC]TELEFONO[SPC]"
720 PRINT"[4CRSRD][5CRSRR]NOMBRE:[4SPC]
[10COMM@]"
730 PRINT"[2CRSRD][5CRSRR]APELLIDOS:
[SPC][20COMM@]"
740 PRINT"[2CRSRD][5CRSRR]TELEFONO:[2SPC]
[7COMM@]"
750 PRINT"[HOM][8CRSRD]":I=0:KK=0:Y=
Y+1:XX=10:VV=32:GOSUB 10000
755 IFB$(0)=CHR$(13)THENY=Y-1:GOTO075
0
760 FOR J=0 TO I-1
770 NOM$(Y)=NOM$(Y)+B$(J)
780 NEXT:N$=B$(0)
790 PRINT"[HOM][11CRSRD]":I=0:KK=120
:XX=20:GOSUB10000
795 IFB$(0)=CHR$(13)THEN790
800 FOR J=0 TO I-1
810 COG$(Y)=COG$(Y)+B$(J)
820 NEXT:C$=B$(0)
830 PRINT"[HOM][14CRSRD]":I=0:KK=240
:XX=7:QL=1:GOSUB10000:QL=0
835 IFB$(0)=CHR$(13)THEN830
840 FOR J=0 TO I-1
850 TEL$(Y)=TEL$(Y)+B$(J)
860 NEXT:T$=B$(0)
870 PRINT"[2CRSRD][2CRSRR][RVSON][SPC]
SI[SPC]SON[SPC]LOS[SPC]DATOS[SPC]COR
RECTOS[SPC]APRIETA[SPC][CRSRD]"
880 PRINT"[RVSON][5CRSRR][SPC]RETURN
[SPC]SI[SPC]SON[SPC]INCORRECTOS[SPC]
(I)[SPC]"
890 GETA$
900 IFA$="I"THENGOSUB2000:GOTO0870
910 IFA$<>CHR$(13)THEN890
980 FORHH=1TOY-1
990 IFCOG$(HH)>COG$(Y)THEN1006
1000 NEXTHH
1003 GOTO1040
1006 RN$=NOM$(Y)
1007 RC$=COG$(Y)
1008 RT$=TEL$(Y)
1010 FORFF=YTOHHSTEP-1
1012 NOM$(FF)=NOM$(FF-1)
1013 COG$(FF)=COG$(FF-1)
1014 TEL$(FF)=TEL$(FF-1)
1016 NEXTFF
1020 NOM$(HH)=RN$
1023 COG$(HH)=RC$
1026 TEL$(HH)=RT$
1040 PRINT"[CRSRD][RVSON][2CRSRR][SPC]
SI[SPC]NO[SPC]ENTRAS[SPC]MAS[SPC]NOB
RES[SPC]APRIETA[SPC](P)[SPC]"
1050 GETA$:IFA$=""THEN1050
1060 IFA$=CHR$(13)THEN700
1070 IFA$="P"THEN:RETURN
1080 GOTO1050
1234 AL MENU GENERAL CON RETURN 1234
56
2000 PRINT"[3CRSRU][40SPC]"
2002 PRINT"[36SPC]"
2005 PRINT"[HOM][8CRSRD]":I=0:KK=0:X
X=10:VV=ASC(N$)-64:UU=1:GOSUB10000
2010 IFB$(0)=CHR$(13)THEN 2050
2015 NOM$(Y)=""
2020 FOR J=0 TO I-1
2030 NOM$(Y)=NOM$(Y)+B$(J)
2040 NEXT:N$=B$(0)
2050 PRINT"[CRSRD]":I=0:KK=120:XX=20
:VV=ASC(C$)-64:UU=1:GOSUB10000
2060 IFB$(0)=CHR$(13)THEN 2100
2065 COG$(Y)=""
2070 FOR J=0 TO I-1
2080 COG$(Y)=COG$(Y)+B$(J)
2090 NEXT:C$=B$(0)
2100 PRINT"[CRSRD]":I=0:KK=240:XX=7:
VV=ASC(T$):UU=1:GOSUB10000
2110 IFB$(0)=CHR$(13)THEN 2150
2115 TEL$(Y)=""
2120 FOR J=0 TO I-1
2130 TEL$(Y)=TEL$(Y)+B$(J)
2140 NEXT:T$=B$(0)
2150 RETURN
3000 PRINT"[CLR][2CRSRD][9CRSRR][RVSON]
[2SPC]AGENDA[SPC]TELEFONICA[2SPC]"
3010 PRINT"[CRSRD][9CRSRR][RVSON][2SPC]
BUSCAR[SPC]UN[SPC]TELEFONO[SPC]"
3020 PRINT"[3CRSRD]ENTRA[SPC]EL[SPC]
APELLIDO:"
3030 PRINT"[6CRSRD][5CRSRR]APELLIDO:
[2SPC][20COMM@]"
3032 PRINT"[6CRSRD][2CRSRR][RVSON][SPC]
PARA[SPC]VER[SPC]TODA[SPC]LA[SPC]LIS
TA[SPC]PULSA[SPC]RETURN[SPC]"
3040 PRINT"[HOM][14CRSRD]":I=0:KK=24
0:XX=20:VV=32:RS=0:GOSUB 10000
3050 PRINT"[CLR][2CRSRD][9CRSRR][RVSON]
[2SPC]AGENDA[SPC]TELEFONICA[2SPC]"
3060 PRINT"[CRSRD][9CRSRR][RVSON][2SPC]
BUSCAR[SPC]UN[SPC]TELEFONO[SPC][CRSRD]
"
3070 IFB$(0)=CHR$(13)THENRR=1
3100 FOR WW=0 TO I-1
3110 IG$=IG$+B$(WW)
3120 NEXT WW
3130 PIG=LEN(IG$)
3135 PP=0
3140 FOR TT=1 TO Y
3145 IFRR=1THEN3152
3150 IFLEFT$(COG$(TT),PIG)<>IG$THEN3
170
3152 PP=PP+1
3160 PRINTTAB(0)COG$(TT);TAB(21)NOM$
(TT);TAB(32)TEL$(TT):RS=1
3170 IF PP<16THEN 3179
3172 PP=0:PRINT"[CRSRD][5CRSRR][RVSON]
[SPC]PARA[SPC]CONTINUAR[SPC]PULSA[SPC]
RETURN[SPC]"
3174 GETA$:IFA$<>CHR$(13)THEN3174
3176 PRINT"[CLR][2CRSRD][9CRSRR][RVSON]
[2SPC]AGENDA[SPC]TELEFONICA[2SPC]"
3177 PRINT"[CRSRD][10CRSRR][RVSON][SPC]
BUSCAR[SPC]UN[SPC]TELEFONO[SPC][CRSRD]
"
3179 NEXT TT
3180 IG$="":RR=0
3185 IFRS=0THENPRINT"[7CRSRD][10SPC]
NOMBRE[SPC]NO[SPC]EXIXTENTE[6CRSRD]"
3190 PRINT"[CRSRD][5CRSRR][RVSON][SPC]
PARA[SPC]CONTINUAR[SPC]PULSA[SPC]RET
URN[SPC]"
3380 GETA$:IFA$<>CHR$(13)THEN3380
3390 RETURN
4000 PRINT"[CLR][2CRSRD][9CRSRR][RVSON]
[2SPC]AGENDA[SPC]TELEFONICA[2SPC]"
4010 PRINT"[CRSRD][9CRSRR][RVSON][2SPC]
BORRAR[SPC]UN[SPC]TELEFONO[SPC]"
4020 PRINT"[3CRSRD]ENTRA[SPC]EL[SPC]
APELLIDO:"
4030 PRINT"[6CRSRD][5CRSRR]APELLIDO:
[2SPC][20COMM@]"
4040 PRINT"[5CRSRD][CRSRR][RVSON][SPC]
```

```
CUIDADO[SPC]CON[SPC]LO[SPC]QUE[SPC]Q
UIERES[SPC]QUE[SPC]BORRE[SPC]"
4045 PRINT"[CRSRD][6CRSRR][RVSON][SPC]
AL[SPC]MENU[SPC]GENERAL[SPC]CON[SPC]
RETURN[SPC]"
4050 PRINT"[HOM][14CRSRD]":I=0:KK=24
0:XX=20:VV=32:GOSUB 10000
4055 IFB$(0)=CHR$(13)THENRETURN
4060 FOR J=0 TO I-1
4070 CAG$=CAG$+B$(J)
4080 NEXT
4085 LF=LEN(CAG$)
4090 PRINT"[CLR][2CRSRD][9CRSRR][RVSON]
[2SPC]AGENDA[SPC]TELEFONICA[2SPC]"
4100 PRINT"[CRSRD][9CRSRR][RVSON][2SPC]
BORRAR[SPC]UN[SPC]TELEFONO[SPC]"
4110 PRINT"[CRSRD][2CRSRR][RVSON][SPC]
ESTE[SPC]ES[SPC]EL[SPC]NOMBRE[SPC]QU
E[SPC]HE[SPC]BORRADO:[SPC][6CRSRD]"
4115 JK=0:RS=0
4120 FOR T=1 TO Y
4130 IFLEFT$(COG$(T),LF)<>CAG$THEN41
60
4140 PRINTTAB(0)COG$(T);TAB(21)NOM$(
T);TAB(32)TEL$(T):RS=1
4150 JK=JK+1
4160 IFJK=0THEN4200
4170 COG$(T)=COG$(T+JK):COG$(T+JK)="
"
4180 NOM$(T)=NOM$(T+JK):NOM$(T+JK)="
"
4190 TEL$(T)=TEL$(T+JK):TEL$(T+JK)="
"
4200 NEXT
4205 IFRS=0THENPRINT"[10SPC]NOMBRE[SPC]
NO[SPC]EXISTENTE"
4210 PRINT"[7CRSRD][5CRSRR][RVSON][SPC]
PULSA[SPC]RETURN[SPC]PARA[SPC]CONTIN
UAR[SPC]"
4230 Y=Y-JK
4240 CAG$="":GETA$:IFA$=""THEN4240
4250 RETURN
5000 PRINT"[CLR][2CRSRD][9CRSRR][RVSON]
[2SPC]AGENDA[SPC]TELEFONICA[2SPC]"
5010 PRINT"[CRSRD][9CRSRR][RVSON][SPC]
GUARDANDO[3SPC]FICHERO[SPC][2CRSRD]"

5020 OPEN1,1,1,"AGENDA"
5030 PRINT#1,Y
5040 FOR T=1 TO Y
5050 PRINT#1,COG$(T)
5060 NEXT
5070 FOR T=1 TO Y
5080 PRINT#1,NOM$(T)
5090 NEXT
5100 FOR T=1 TO Y
5110 PRINT#1,TEL$(T)
5120 NEXT
5130 CLOSE1
5140 RETURN
6000 OPEN1,1,0,"AGENDA"
6070 INPUT#1,Y
6080 FOR T=1 TO Y
6090 INPUT#1,COG$(T)
6100 NEXT
6110 FOR T=1 TO Y
6120 INPUT#1,NOM$(T)
6130 NEXT
6140 FOR T=1 TO Y
6150 INPUT#1,TEL$(T)
6160 NEXT
6170 CLOSE1
6180 RETURN
10000 FOR T=1 TO 30
10100 GETB$(I):IFB$(I)=""THEN NEXT T

10200 IF B$(I)=CHR$(13)THENPOKE1400+
I+KK,VV:PRINT:RETURN
10210 IFB$(I)=CHR$(20)THENI=I-2:GOSU
B12200
10220 IFI<0THENI=0:GOTO10000
10250 IFB$(0)<>""ANDUU=1THENGOSUB120
00
10300 IFB$(I)<>""THEN 10450
10400 POKE1400+I+KK,VV+G:GOTO 10700
10450 IFB$(1)<>""THENVV=32
10460 MM=ASC(B$(I))
10470 IFMM=46ORMM=32ORMM>47ANDMM<58T
HENK=0:GOTO10500
10480 IFMM>64ANDMM<91THENK=64:GOTO10
500
10490 GOTO10000
10500 POKE1400+I+KK,MM-K
10600 IFI=XXTHEN10000
10650 I=I+1
10700 IF G=0 THEN G=128:GOTO10000
10800 IF G=128 THEN G=0:GOTO10000
10900 GETA$:IFA$<>CHR$(13)THEN10900
11000 PRINT:RETURN
12000 FORU=0TOXX-1:POKE1400+KK+U,100
:NEXT:UU=0:RETURN
12200 POKE1402+I+KK,100:RETURN
13000 PRINT"[CLR]"
13010 FORS=1TO164
13020 READQ:POKE1024+Q,224
13030 NEXTS
13040 PRINT"[14CRSRD][8CRSRR]PROGRAM
A[SPC]REALITZADO[SPC]POR:"
13050 PRINT"[2CRSRD][7CRSRR]*******
****************"
13060 PRINT"[7CRSRR]**[21SPC]**"
13070 PRINT"[7CRSRR]**[SPC]JORDI[SPC]
CARDONA[SPC]VIDAL[SPC]**"
13080 PRINT"[7CRSRR]**[21SPC]**"
13090 PRINT"[7CRSRR]***************
********"
13100 FORS=1TO8000:NEXT:RETURN
13140 DATA88,89,90,92,93,94,96,97,98
,100,103,105,106,109,110,111
13150 DATA128,130,132,136,140,141,14
3,145,147,149,151
13160 DATA168,169,170,172,173,174,17
6,177,180,181,182,183,185,187,189,19
0,191
13170 DATA208,210,212,214,216,220,22
2,223,225,227,229,231
13180 DATA248,250,252,253,254,256,25
7,258,260,263,265,266,269,271
13190 DATA321,322,323,325,326,327,32
9,333,334,335,337,338,339,341,342,34
3,345
13200 DATA348,350,352,353,354,356,35
7,358
13210 DATA362,365,369,373,377,381,38
3,385,386,388,390,392,396,398
13220 DATA402,405,406,409,413,414,41
7,418,421,423,425,426,427,428,430,43
2,436
13230 DATA437,438,442,445,449,453,45
7,461,463,465,467,468,470,472,476,47
8
13240 DATA482,485,486,487,489,490,49
1,493,494,495,497,501,502,503,505,50
8,510
13250 DATA512,513,514,516,518
```



Comando auto

José Ramón Gana
Gral. Castaños, nº 47-1º dcha.
Portugalete (Vizcaya)

VIC-20

Os envío una rutina que simula el funcionamiento del comando AUTO.

Este comando es muy útil para teclear programas en los que los números de línea de la mayoría de las instrucciones siguen un incremento constante. Para introducir las instrucciones que no siguen la norma general del programa que estamos metiendo basta con parar la ejecución de la rutina y teclearlas normalmente.

Esta rutina tiene unos números de línea muy grandes para dejar margen al programa que se va a introducir.

Una vez que hemos acabado de utilizar la rutina, ejecutando a partir de la instrucción 60400 se consigue borrar la rutina y dejar en memoria sólo el programa que nos interesa.

La rutina funciona con ampliaciones de 8K ó 16K. En caso de no estar conectadas estas ampliaciones basta con cambiar:

Línea 60020:

El número 4140 por 7724
4514 por 8098

Línea 60070:

4162 por 7746
4514 por 8098

Estos números afectan a la memoria de pantalla.

La rutina lista las últimas instrucciones que hemos tecleado pero para modificar estas instrucciones hay que salirse de la ejecución de la rutina. Es decir, una vez que se está ejecutando la rutina sólo se puede actuar sobre una instrucción. Espero que esta rutina os sirva de algo.

```
60000 PRINT"[CLR][6CRSRD]DA[SPC]EL[SPC]
NUMERO[SPC]INICIAL[2CRSRD][2SPC]Y[SPC]
EL[SPC]INCREMENTO[2CRSRD]"
60010 INPUTNI,IN:PRINT"[CLR]":POKE90
5,0
60020 FORI=4140TO4514STEP22
60030 IFPEEK(I)<>18ORPEEK(I+1)<>5ORP
EEK(I+2)<>1ORPEEK(I+3)<>4ORPEEK(I+4)
<>25THEN60050
60040 FORP=ITOI+5:POKEP,32:NEXT
60050 NEXT
60060 S$="[3CRSRD]"
60070 FORI=4162TO4514STEP22
60080 FORJ=ITOI+22
60090 IFPEEK(J)<>32THENS$=S$+"[CRSRD]
":J=I+22:GOTO60110
60100 NEXT:GOTO60120
60110 NEXT:NEXT
60120 PRINT"[HOM]"S$"[CRSRL][BLU]";N
I;:B$=STR$(NI)
60130 POKE204,0
60140 GETA$:IFA$=""THEN60140
60150 IFA$=CHR$(13)THENPRINT:GOTO601
80
60160 PRINTA$;:B$=B$+A$
60170 GOTO60140
60180 IFI<60000THEN60200
60190 PRINT"[CLR][BLU]NUMERO[SPC]DE[SPC]
LINEA[7SPC]MUY[SPC]GRANDE":FORM=1TO5
000:NEXT:GOTO60000
60200 PRINT"[CLR][WHT]";B$:PRINT"GOT
O60250"
60210 POKE198,3:POKE631,19:POKE632,1
3:POKE633,13
60220 C1=INT(NI/256):R1=NI-C1*256:C2
=INT(IN/256):R2=IN-C2*256
60230 POKE900,C1:POKE901,R1:POKE902,
C2:POKE903,R2:POKE904,1
60240 END
60250 NI=PEEK(900)*256+PEEK(901)
60260 IN=PEEK(902)*256+PEEK(903)
60270 IFPEEK(904)=0THENNI=NI+IN:GOTO
60020
60280 PRINT"[CLR]LIST"NI-IN*PEEK(905
)"-"NI:PRINT"GOTO60250":PRINT"[BLU]"

60290 IFPEEK(905)<3THENPOKE905,PEEK(
905)+1
60300 POKE198,5:POKE631,19:POKE632,1
3:POKE633,19:POKE634,17:POKE635,13:P
OKE904,0
60310 END
60400 N=60000:K=8
60410 Z=K:PRINT"[CLR][WHT]";:FORM=NT
ON+40STEP10:PRINTM:NEXT
60420 IFN>60380THENK=5:Z=8:POKE637,1
47:POKE638,31:GOTO60440
60430 PRINT"60400[SPC]N="M":K="8:PRI
NT"RUN[SPC]60400"
60440 POKE198,Z:POKE631,19:FORI=1TOK
:POKE631+I,13:NEXT
```



La hora del programa

Para saber a qué hora has guardado un programa, escribe al hacer SAVE...

SAVE "nombre"+ti$

Has de tener en cuenta que el nombre no puede tener más de 10 caracteres, y previamente tienes que haber asignado a ti$ la hora correcta.

L.F.S. RUN EE.UU.

Nombres mágicos

Prueba a salvar un programa en la unidad de discos con el nombre:

CHR$(0)+"NOMBRE DEL PROGRAMA"

Ahora el programa sólo podrá cargarse de esta manera.

B.B. RUN EE.UU.

Comunidades

Antonio Luis Climent Albaladejo
Comuneros, 1-2º B
30003 MURCIA

C-64



Las reglas del juego vienen suficientemente explicadas en las instrucciones correspondientes a las líneas 940-1160, visualizándose el contenido de las sentencias Print a continuación de la presentación. Debe resaltarse que las condiciones de partida de cada una de las 17 Comunidades Autónomas es idéntica (población, riqueza, cultura y empleo).
La estrategia del juego consiste en buscar la distribución óptima de nuestros votantes entre las distintas regiones a fin de ganar las elecciones en el mayor número de ellas.
El mapa está realizado con gráficos de baja resolución, por lo que son apreciables algunos errores y omisiones de accidentes geográficos.
La música no es original. Se trata de una adaptación efectuada por Pepe Bermejo (mi cuñado) sobre temas regionales y sobre la musiquilla del "telediario".
Realizando unas pruebas sobre este programa me comentaron que para seguir el desarrollo de la partida casi hacía falta papel y lápiz. Mi respuesta fue que una forma de evitar papel y lápiz para efectuar complejos cálculos estratégicos era conveniente utilizar la opción "¿cuántos mueve?"=0, con lo que se anula el movimiento no sin antes observar el número y distribución de votantes en la región de origen y en la de destino.
El juego finaliza una vez superado el tiempo prefijado de juego. Exactamente, una vez celebrado el primer comicio electoral posterior al tiempo de juego fijado al inicio de la partida.

```
10 VT$="[HOM][29CRSRD]":PRINT"[BLK]"
:GOSUB 2910
20 GOSUB 930:REM        INSTRUCCIONES
30 GOSUB1190:REM JUGADORES
40 GOSUB 1270:REM   ESTABLECE CONDIC
IONES INICIALES Y TIEMPO JUEGO
50 PRINT"[CLR]":GOSUB 2270 :REM  MOV
 Y ELECCIONES
60 M$="[12SPC]TIEMPO:[3SPC]"+TI$+"[10SPC]
":GOSUB 2440
70 IFTI$<TJ$THEN 50
80 FOR I=1TO 17 :PRINT,R$(I),PG$(I)
90 FOR J=1TONJ:IF PG$(I)=P$(J)THEN P
(J)=P(J)+1
```

```
100 NEXT J,I
110 MA=-1 :FOR J=1 TO NJ: IF P(J)>MA
THEN MA=P(J):I=J
120 NEXT J
130 M$="[15SPC]GANADOR[2SPC]"+N$(I)+
"[10SPC]":GOSUB 3690:GOSUB 2440
140 END
150 GOTO 150
160 REM GALICIA
170 PRINTLEFT$(VT$,3)TAB(9);
180 PRINT"[COMM3][RVSON][SHIFT£][COMM*]
[3CRSRL][CRSRD][SHIFT£][3SPC][4CRSRL]
[CRSRD][5SPC][5CRSRL][CRSRD][RVSOFF]
[COMM*][RVSON][3SPC][4CRSRL][CRSRD][SHIFT£]
[3SPC]"
190 RETURN
200 REM ASTURIAS
210 PRINTLEFT$(VT$,4)TAB(12);
220 PRINT  "[BLK][RVSON][4SPC][3CRSRL]
[CRSRD][2SPC]"
230 RETURN
240 REM CANTABRIA
250 PRINTLEFT$(VT$,4)TAB(16);
260 PRINT"[COMM3][RVSON][2SPC][3CRSRL]
[CRSRD][3SPC][2CRSRL][CRSRD][SPC]"
270 RETURN
280 REM PAIS VASCO
290 PRINTLEFT$(VT$,3)TAB(18)
300 PRINT"[BLK][RVSON][SHIFT£][COMM*]
[2CRSRL][CRSRD][4SPC][4CRSRL][CRSRD]
[4SPC][3CRSRL][CRSRD][2SPC]"
310 RETURN
320 REM FRANCIA
330 PRINTLEFT$(VT$,3)TAB(22);
340 PRINT"[BLK][RVSON][9SPC][RVSOFF]
[SHIFT£][9CRSRL][CRSRD][RVSON][7SPC]
[RVSOFF][SHIFT£][5CRSRL][CRSRD][RVSON]
[4SPC][COMM*]"
350 RETURN
360 REM NAVARRA
370 PRINTLEFT$(VT$,4)TAB(22);
380 PRINT"[COMM6][RVSON][SPC][CRSRL]
[CRSRD][2SPC][3CRSRL][CRSRD][3SPC][2CRSRL]
[CRSRD][SPC]"
390 RETURN
400 REM ARAGON
410 PRINTLEFT$(VT$,5)TAB(24);
420 PRINT"[COMM3][RVSON][2SPC][2CRSRL]
[CRSRD][2SPC][3CRSRL][CRSRD][3SPC][4CRSRL]
[CRSRD][4SPC][4CRSRL][CRSRD][4SPC][5CRSRL]
[CRSRD][5SPC]";
430 PRINT"[RVSON][5CRSRL][CRSRD][4SPC]
[3CRSRL][CRSRD][2SPC][CRSRL][CRSRD][SPC]
":RETURN
440 REM CATALUNA
450 PRINTLEFT$(VT$,5)TAB(26);
460 PRINT"[COMM6][RVSON][SPC][CRSRL]
[CRSRD][5SPC][5CRSRL][CRSRD][5SPC][5CRSRL]
[CRSRD][4SPC][RVSOFF][SHIFT£]";
470 PRINT"[RVSON][5CRSRL][CRSRD][2SPC]
[RVSOFF][SHIFT£][3CRSRL][CRSRD][RVSON]
[SPC][RVSOFF][SHIFT£]":RETURN
480 REM CASTILLA LEON
490 PRINTLEFT$(VT$,6)TAB(12);
500 PRINT"[COMM6][RVSON][4SPC][CRSRR]
[2SPC][7CRSRL][CRSRD][7SPC][7CRSRL][CRSRD]
[6SPC]";
510 PRINT"[RVSON][5CRSRL][CRSRD][5SPC]
[CRSRR][3SPC][9CRSRL][CRSRD][8SPC][9CRSRL]
[CRSRD][9SPC]";
520 PRINT"[RVSON][9CRSRL][CRSRD][6SPC]
[6CRSRL][CRSRD][SPC][CRSRR][3SPC]";:
RETURN
530 REM RIOJA
540 PRINTLEFT$(VT$,7)TAB(19);
550 PRINT"[WHT][RVSON][3SPC][4CRSRL]
[CRSRD][4SPC][4CRSRL][CRSRD][SPC]":R
ETURN
560 REM PORTUGAL
```

```
570 PRINTLEFT$(VT$,8)TAB(8);
580 PRINT"[BLK][RVSON][4SPC][4CRSRL]
[CRSRD][5SPC][5CRSRL][CRSRD][5SPC][5CRSRL]
[CRSRD][4SPC][5CRSRL][CRSRD]";
590 PRINT"[RVSON][SHIFT£][4SPC][5CRSRL]
[CRSRD][5SPC][5CRSRL][CRSRD][5SPC][5CRSRL]
[CRSRD][4SPC][3CRSRL][CRSRD][3SPC]";

600 PRINT"[RVSON][3CRSRL][CRSRD][4SPC]
[5CRSRL][CRSRD][SHIFT£][3SPC][4CRSRL]
[CRSRD][3SPC]"
610 RETURN
620 REM EXTREMADURA
630 PRINTLEFT$(VT$,13)TAB(13);
640 PRINT"[WHT][RVSON][SPC][2CRSRL][CRSRD]
[3SPC][4CRSRL][CRSRD][5SPC][5CRSRL][CRSRD]
[4SPC][3CRSRL][CRSRD][2SPC]":RETURN
650 REM MADRID
660 PRINTLEFT$(VT$,12)TAB(18);
670 PRINT"[COMM3][RVSON][SPC][2CRSRL]
[CRSRD][3SPC]":RETURN
680 REM CASTILLA-LA MANCHA
690 PRINTLEFT$(VT$,12)TAB(19);
700 PRINT"[BLK][RVSON][3SPC][2CRSRL]
[CRSRD][3SPC][8CRSRL][CRSRD][9SPC][8CRSRL]
[CRSRD]";
710 PRINT"[RVSON][7SPC][8CRSRL][CRSRD]
[7SPC][6CRSRL][CRSRD][5SPC]":RETURN
720 REM VALENCIA
730 PRINTLEFT$(VT$,11)TAB(25);
740 PRINT"[WHT][RVSON][SPC][RVSOFF][SHIFT£]
[3CRSRL][CRSRD][RVSON][SPC][RVSOFF][SHIFT£]
[2CRSRL][CRSRD][RVSON][SPC][COMM*][2CRSRL]
[CRSRD][2SPC][3CRSRL][CRSRD][2SPC][RVSOFF]
[SHIFT£][3CRSRL][CRSRD][RVSON][SPC][RVSOFF]
[SHIFT£][2CRSRL][CRSRD][SHIFT£]"
750 RETURN
760 REM BALEARES
770 PRINTLEFT$(VT$,12)TAB(29);
780 PRINT"[BLK][RVSON][SHIFT£][CRSRL]
[CRSRD][RVSOFF][COMM*][SHIFT£][CRSRR]
[2CRSRU][COMMO][2CRSRD][4CRSRL][2CRSRD]
[CRSRL][SHIFTQ]":RETURN
790 REM MURCIA
800 PRINTLEFT$(VT$,15)TAB(22);
810 PRINT"[COMM3][RVSON][SPC][2CRSRL]
[CRSRD][2SPC][3CRSRL][CRSRD][3SPC][2CRSRL]
[CRSRD][2SPC]":RETURN
820 REM ANDALUCIA
830 PRINTLEFT$(VT$,17)TAB(14);
840 PRINT"[COMM6][RVSON][2SPC][5CRSRL]
[CRSRD][10SPC][11CRSRL][CRSRD][11SPC]
[RVSOFF][SHIFT£]";
850 PRINT"[11CRSRL][CRSRD][RVSOFF][COMM*]
[RVSON][9SPC][9CRSRL][CRSRD][RVSOFF]
[COMM*][RVSON][2SPC][RVSOFF][SHIFT£]
[2CRSRR][COMM*][SHIFT£][7CRSRL][CRSRD]
[RVSON][SPC][RVSOFF][SHIFT£]";
860 RETURN
870 REM CANARIAS
880 PRINTLEFT$(VT$,20)TAB(27)"[SHIFTO]
[12COMMY][13CRSRL][CRSRD][COMMG][CRSRD]
[CRSRL][COMMG][CRSRD][CRSRL][COMMG][CRSRD]
[CRSRL][COMMG][CRSRD][CRSRL][COMMG][CRSRU]
";
890 PRINTLEFT$(VT$,23)TAB(29);
900 PRINT"[BLK][SHIFTQ][CRSRR][COMM*]
[SHIFT£][CRSRL][CRSRU][COMMO][COMMF]
[CRSRR][SHIFTQ][CRSRR][CRSRD][RVSON]
[COMMK][CRSRL][CRSRU][COMMJ][RVSOFF]
[CRSRU][SHIFTQ]"
910 RETURN
920 FORI=1TO5000:NEXTI:RETURN
930 PRINT"[COMM1]";
940 PRINT"[CLR][CRSRD]EL[SPC]TERRITO
RIO[SPC]ESPANOL[SPC]ESTA[SPC]ORGANIZ
ADO[SPC]EN17[SPC]COMUNIDADES[SPC]AUT
ONOMAS."
950 PRINT"[CRSRD]AL[SPC]INICIO[SPC]D
EL[SPC]JUEGO[SPC]TODAS[SPC]LAS[SPC]R
EGIONES[2SPC]TIENEN[SPC]LA[SPC]MISMA
[SPC]";
960 PRINT"CAPACIDAD[SPC]DE[SPC]DESAR
ROLLO[SPC]SOCIOECONOMICO[SPC]";
970 PRINT"QUE[SPC]SE[SPC]ALTERARA[2SPC]
DURANTE[SPC]EL[SPC]CURSO[SPC]DE[SPC]
LA[SPC]PARTIDA."
980 PRINT"[CRSRD]EL[SPC]OBJETIVO[SPC]
DE[SPC]CADA[SPC]JUGADOR[SPC]ES[SPC]O
BTENER[2SPC]EL[SPC]GOBIERNO[SPC]EN[SPC]
EL";
990 PRINT"[SPC]MAYOR[SPC]NUMERO[SPC]
DE[SPC]REGIO-NES."
1000 PRINT"[CRSRD]INICIALMENTE[SPC]S
E[SPC]INTRODUCEN[SPC]LOS[SPC]NOMBRES
[2SPC]DE[SPC]LOS[SPC]JUGADORES[SPC]"
;
1010 PRINT"Y[SPC]EL[SPC]TIEMPO[SPC]D
E[SPC]JUEGO."
1020 PRINT"[CRSRD]LA[SPC]PARTIDA[SPC]
SE[SPC]DESARROLLA[SPC]DE[SPC]LA[SPC]
SIGUIENTEFORMA:":GOSUB 920
1030 PRINT"[CRSRD]CADA[SPC]JUGADOR[SPC]
POR[SPC]TURNO[SPC]DE[SPC]JUEGO[SPC]P
UEDE[3SPC]ORDENAR[SPC]UN[SPC]MOVIMIE
NTO[SPC]";
1040 PRINT"MIGRATORIO[SPC]DE[SPC]RE-
[SPC]GION[SPC]A[SPC]REGION[SPC]TRASL
ADANDO,[SPC]DE[SPC]ESTA[SPC]FORMALOS
";
1050 PRINT"[SPC]VOTOS[SPC]DE[SPC]SU[SPC]
PARTIDO[SPC]AL[SPC]OBJETO[SPC]DE[SPC]
GA-NAR[SPC]LAS[SPC]ELECCIONES."
1060 GOSUB 920:PRINT"[CRSRD]DESPUES[SPC]
DE[SPC]CADA[SPC]MOVIMIENTO[SPC]MIGRA
TORIO[3SPC]TIENE[SPC]LUGAR[SPC]";
1070 PRINT"UN[SPC]ACONTECIMIENTO[SPC]
ALEATORIO[SPC]QUE[SPC]AFECTARA[SPC]A
[SPC]LAS[SPC]CONDICIONES[SPC]";
1080 PRINT"INICIALESDE[SPC]CADA[SPC]
REGION[SPC]Y[SPC]A[SPC]LOS[SPC]VOTOS
[SPC]DE[SPC]LOS[SPC]PAR-TIDOS[SPC]QU
E[SPC]";
1090 PRINT"ESTAN[SPC]PRESENTES[SPC]E
N[SPC]LAS[SPC]REGIO-[SPC]NES[SPC]AFE
CTADAS."
1100 GOSUB 920:PRINT"[CRSRD]CUANDO[SPC]
HAN[SPC]JUGADO[SPC]TODOS[SPC]LOS[SPC]
PARTIDOS[4SPC]";
1110 PRINT"DURANTE[SPC]CUATRO[SPC]AN
OS[SPC](CUATRO[SPC]MOVIMIENTOS)";
1120 PRINT"TIENEN[SPC]LUGAR[SPC]LAS[SPC]
NUEVAS[SPC]ELECCIONES[SPC]QUE[2SPC]"
1130 PRINT"ESTABLECERAN[SPC]NUEVOS[SPC]
GOBIERNOS[SPC]EN[SPC]LAS[4SPC]REGION
ES[SPC]AUTONOMAS."
1140 GOSUB 920:PRINT"[CRSRD]SE[SPC]R
EPITE[SPC]ESTA[SPC]SECUENCIA[SPC]HAS
TA[SPC]AGOTAR[SPC]EL"
1150 PRINT"TIEMPO[SPC]DE[SPC]JUEGO[SPC]
PREFIJADO[2SPC]"
1160 PRINT"[2CRSRD][RVSON][6SPC]PULS
E[SPC]UNA[SPC]TECLA[SPC]PARA[SPC]CON
TINUAR[4SPC]"
1170 GET RE$:IF RE$=""THEN 1170
1180 RETURN
1190 INPUT"[CLR][3CRSRD]NUMERO[SPC]D
E[SPC]JUGADORES[3SPC]3[3CRSRL]";RE$
1200 IF RE$<"2"ORRE$>"9"THEN 1190
1210 NJ=VAL(RE$):FOR I=1 TO NJ:PRINT
"[2CRSRD]NOMBRE[SPC]JUGADOR";I:INPUT
 N$(I)
1220 INPUT"[CRSRD]SIGLAS[SPC]PARTIDO
";P$(I):PRINT,,P$(I):P$(I)=LEFT$(P$(
I),7):NEXT I
1230 RETURN
1240 INPUT "TIEMPO[SPC]DE[SPC]JUEGO[SPC]
```

```
(HHMMSS)[SPC]010000[8CRSRL]";RE$
1250 TI$="000000":TJ$=RE$
1260 RETURN
1270 DIM CR(17,3),V(NJ,17),R$(17),R(
20),PG$(17)
1280 FOR I=1 TO 17: FOR J=1TO 3:CR(I
,J)=100:NEXT J,I:DIM SG$(39),CG(39,3
)
1290 FOR I=1 TO 19:READ SG$(I),CG(I,
1),CG(I,2),CG(I,3):NEXT I
1300 FOR I=1 TO 20:READ SG$(19+I),CG
(19+I,1),CG(19+I,2),CG(19+I,3):NEXT
I
1310 FOR I=1 TO 17:READ R$(I):NEXT I

1320 FOR I=1 TO 17: FOR J=1TO NJ:V(J
,I)=INT(16000000/17/NJ):NEXT J,I
1330 FOR I=1 TO 20:READ R(I):NEXT I
1340 GOSUB 1240
1350 RETURN
1360 FOR J=1 TO NJ:PRINT"[BLK][CLR][2CRSRD]
JUEGA[SPC]";N$(J)
1370 FOR I=1 TO 17:PRINT ,I TAB(15),
R$(I):NEXT I
1380 INPUT "[2CRSRD]REGION[SPC]ORIGE
N[SPC]MOVIMIENTO[SPC]MIGRATORIO";RE$
:O=VAL(RE$)
1390 IF RE$<"1"ORRE$>"9"THEN1380
1400 INPUT "REGION[SPC]DESTINO[SPC]M
OVIMIENTO[SPC]MIGRATORIO";RE$
1410 IF RE$<"1"ORRE$>"9"THEN1400
1420 D=VAL(RE$):IF O<1ORD<1ORO>17ORD
>17OR O=D THEN 1360
1430 PRINT"[CLR][CRSRD][RVSON]ORIGEN
:[RVSOFF][CRSRD][5CRSRR]"PG$(O);
1440 IFO<9THENON O GOSUB 180,220,260
,300,380,420,460,500
1450 IFO>8ANDO<16THENON O-8 GOSUB 55
0,670,700,740,780,640,840
1460 IFO>15THENON O-15 GOSUB 810,900

1470 PRINT"[BLK][HOM][2CRSRD]":FOR I
=1 TO NJ
1480 PRINT TAB(25)P$(I)TAB(30)V(I,O)

1490 NEXT I
1500 PRINT"[HOM][10CRSRD][RVSON]DEST
INO[RVSOFF][CRSRD][5CRSRR]"PG$(D);
1510 IFD<9THENON D GOSUB 180,220,260
,300,380,420,460,500
1520 IFD>8ANDD<16THENOND-8GOSUB 550,
670,700,740,780,640,840
1530 IFD>15THENOND-15GOSUB 810,900EA
DY.
1540 PRINT"[BLK][HOM][12CRSRD]":FOR
I=1 TO NJ
1550 PRINT TAB(25)P$(I)TAB(30)V(I,D)

1560 NEXT I
1570 INPUT "[4CRSRD]CUANTOS[SPC]MUEV
E[SPC](0 ANULAR)";RE$
1580 IF RE$<"0"OR RE$>"99999999999"T
HEN 1570
1590 M=VAL(RE$):IF M>V(J,O)OR M=0THE
N 1360
1600 REM **** ACEPTADO MOV ****
1610 V(J,O)=V(J,O)-M:V(J,D)=V(J,D)+M

1620 REM **** GENERACION DEL SUCESO
***
1630 M=INT(1+39*RND(0))
1640 R=INT(1+17*RND(0))
1650 IF M>19 THEN R=R(M-19):M$=SG$(M
):GOTO 1670
1660 M$= R$(R)+":[SPC]"+SG$(M)
1670 PRINT"[CLR][4CRSRD][RVSON]NOTIC
IAS[8CRSRL][2CRSRD]"+M$:GOSUB 3690
1680 MA=-1:NM=0
1690 FOR I=1 TO NJ:IF V(I,R)>=MA THE
N MA=V(I,R):GB=I:NM=NM+1:IF NM=1THEN
M1=MA
1700 NEXT I:PG$(R)=P$(GB):IF(NM=NJ)A
ND(MA=M1)THEN1860
1710 NE=INT((CG(M,2)/100+1)*CR(R,2))

1720 NC=INT((CG(M,3)/100+1)*CR(R,3))

1730 NI=INT((CG(M,1)/100+1)*CR(R,1))

1740 NS=NE+NC+NI
1750 NA=CR(R,1)+CR(R,2)+CR(R,3)
1760 EV = INT((NS-NA)/NA*100):REM PO
RCENTAJE GANDO O PERDIDO
1770 CR(R,1)=NI:CR(R,2)=NE:CR(R,3)=N
C
1780 SV=0:FOR I=1 TO NJ:IFI<>GBTHENS
V=SV+V(I,R)
1790 NEXT I
1800 VG=INT(EV*V(GB,R)/100):IF VG>SV
THENVG=SV
1810 VC=-INT(VG/(NJ-1))
1820 FOR I=1 TO NJ
1830 IFI=GBTHEN V(GB,R)=V(GB,R)+ VG:
GOTO 1850
1840 V(I,R)=V(I,R)+ VC
1850 NEXT I
1860 PRINT"[HOM][10CRSRD][RVSON]REGI
ON[SPC]AFECTADA[RVSOFF][CRSRD][5CRSRR]
"PG$(R);
1870 IFR<9THENON R GOSUB 180,220,260
,300,380,420,460,500
1880 IFR>8ANDR<16THENON R-8 GOSUB 55
0,670,700,740,780,640,840
1890 IFR>15THENON R-15 GOSUB 810,900

1900 PRINT"[BLK][HOM][12CRSRD]":FOR
I=1 TO NJ
1910 PRINT TAB(25)P$(I)TAB(30)V(I,R)

1920 NEXT I
1930 GOSUB 920:NEXT J
1940 RETURN
1950 GOSUB 160
1960 GOSUB 200
1970 GOSUB 240
1980 GOSUB 280
1990 GOSUB 320
2000 GOSUB 360
2010 GOSUB 400
2020 GOSUB 440
2030 GOSUB 480
2040 GOSUB 530
2050 GOSUB 560
2060 GOSUB 620
2070 GOSUB 650
2080 GOSUB 680
2090 GOSUB 720
2100 GOSUB 760
2110 GOSUB 790
2120 GOSUB 820
2130 GOSUB 870
2140 RETURN
2150 PRINT"[CLR]":GOSUB 1950:FOR L=1
 TO 17:MA=-1:NM=0
2160 M$="[32SPC]"+R$(L)+":[2SPC]JORN
ADA[SPC]ELECTORAL"
2170 GOSUB 2440
2180 FOR K=1TO NJ
2190 IF V(K,L)>=MATHENMA=V(K,L):GB=K
:NM=NM+1:IFNM=1THEN M1=MA
2200 NEXT K:PG$(L)=P$(GB):IF(NM=NJ)A
ND(MA=M1)THEN PG$(L)="":GOTO 2260
2210 FOR K=1 TO NJ
2220 M$="[37SPC]"+P$(K)+STR$(V(K,L))
:GOSUB 2440
2230 NEXT K
2240 M$="[34SPC]GANA[SPC]ELECCIONES[SPC]
"+PG$(L)+"[9SPC]"
```

```
2250 PRINT"[HOM][CRSRD][16SPC]":GOSU
B 3690:GOSUB2440
2260 NEXT L:RETURN
2270 FOR MV=1 TO 4:GOSUB 1360:NEXT M
V
2280 GOSUB 2150
2290 RETURN
2300 PRINTLEFT$(VT$,5)TAB(26);
2310 PRINT"[BLU][RVSON][SPC][CRSRL][CRSRD]
[5SPC][5CRSRL][CRSRD][5SPC][5CRSRL][CRSRD]
[4SPC][RVSOFF][SHIFT£]";
2320 PRINT"[RVSON][5CRSRL][CRSRD][2SPC]
[RVSOFF][SHIFT£][3CRSRL][CRSRD][RVSON]
[SPC][RVSOFF][SHIFT£]":RETURN
2330 PRINTLEFT$(VT$,20)TAB(27)"[SHIFTO]
[12COMMY][13CRSRL][CRSRD][COMMG][CRSRD]
[CRSRL][COMMG][CRSRD][CRSRL][COMMG][CRSRD]
[CRSRL][COMMG][CRSRD][CRSRL][COMMG][CRSRU]
";
2340 PRINTLEFT$(VT$,23)TAB(29);
2350 PRINT"[BLU][SHIFTQ][CRSRR][COMM*]
[SHIFT£][CRSRL][CRSRU][COMMO][COMMF]
[CRSRR][SHIFTQ][CRSRR][CRSRD][RVSON]
[COMMK][CRSRL][CRSRU][COMMJ][RVSOFF]
[CRSRU][SHIFTQ]"
2360 RETURN
2370 PRINTLEFT$(VT$,3)TAB(18)
2380 PRINT"[BLU][RVSON][SHIFT£][COMM*]
[2CRSRL][CRSRD][4SPC][4CRSRL][CRSRD]
[4SPC][3CRSRL][CRSRD][2SPC]"
2390 RETURN
2400 PRINTLEFT$(VT$,17)TAB(14);
2410 PRINT"[BLU][RVSON][2SPC][5CRSRL]
[CRSRD][10SPC][11CRSRL][CRSRD][11SPC]
[RVSOFF][SHIFT£]";
2420 PRINT"[11CRSRL][CRSRD][RVSOFF][COMM*]
[RVSON][9SPC][9CRSRL][CRSRD][RVSOFF]
[COMM*][RVSON][2SPC][RVSOFF][SHIFT£]
[2CRSRR][COMM*][SHIFT£][7CRSRL][CRSRD]
[RVSON][SPC][RVSOFF][SHIFT£]";
2430 RETURN
2440 FOR I=1 TO 40
2450 PRINT LEFT$(VT$,2)LEFT$(M$,40)
2460 M$=RIGHT$(M$,LEN(M$)-1)+LEFT$(M
$,1)
2470 FOR T=1 TO 20:NEXT T,I:RETURN
2480 DATA "INVERSION[SPC]EXTRANJERA"
,10,10,1
2490 DATA "ESCUELA",5,3,10
2500 DATA "POLIDEPORTIVOS",7,1,4
2510 DATA "EQUIPAMIENTO[SPC]COLECTIV
O[SPC]COMER--CIAL",7,8,1
2520 DATA "HOSPITAL",8,7,0
2530 DATA "CARRETERA",10,5,0
2540 DATA "AUTOPISTA",15,6,0
2550 DATA "ESTABLECIMIENTO[SPC]EMPRE
SA[SPC]PUBLI-CA",8,10,0
2560 DATA "NUEVO[SPC]HOTEL",6,5,1
2570 DATA "ATENTADO[SPC]TERRORISTA",
-1,0,0
2580 DATA "DESCUBIERTO[SPC]TRAFICO[SPC]
DE[SPC]DROGAS",-1,0,-1
2590 DATA "ESTABLECIMIENTO[SPC]DE[SPC]
NUEVA[SPC]LINE-A[SPC]DE[SPC]FF.CC.",
15,16,0
2600 DATA "ESTABLECIMIENTO[SPC]DE[SPC]
NUEVO[SPC]AERO-PUERTO",20,8,0
2610 DATA "AUMENTAN[SPC]DELITOS[SPC]
COMUNES",-1,-1,0
2620 DATA "ESCANDALO[SPC]POLITICO",0
,-1,-2
2630 DATA "CRISIS[SPC]PARTIDO",0,0,0

2640 DATA "CIERRE[SPC]EMPRESA",-10,-
10,0
2650 DATA "CIERRE[SPC]EMPRESA",-3,-3
,0
2660 DATA "CIERRE[SPC]EMPRESA",-8,-8
,0
2670 DATA"CASTILLA[SPC]LEON:[SPC]BUE
NA[SPC]COSECHA",1,7,0
2680 DATA"CASTILLA[SPC]LA[SPC]MANCHA
:[SPC]BUENA[SPC]COSE-CHA",1,7,0
2690 DATA"RIOJA:[SPC]BUENA[SPC]COSEC
HA",1,7,0
2700 DATA"MURCIA:[SPC]BUENA[SPC]COSE
CHA",1,7,0
2710 DATA"VALENCIA:[SPC]BUENA[SPC]CO
SECHA",1,7,0
2720 DATA"ANDALUCIA:[SPC]BUENA[SPC]C
OSECHA",1,7,0
2730 DATA"PAIS[SPC]VASCO:[SPC]PARO[SPC]
GENERAL",0,-1,0
2740 DATA"ASTURIAS:[SPC]PARO[SPC]GEN
ERAL",0,-1,0
2750 DATA"ASTURIAS:[SPC]AUMENTA[SPC]
TURISMO",1,3,0
2760 DATA"CANTABRIA:[SPC]AUMENTA[SPC]
TURISMO",1,3,0
2770 DATA"CATALUNA:[SPC]AUMENTA[SPC]
TURISMO",1,3,0
2780 DATA"VALENCIA:[SPC]AUMENTA[SPC]
TURISMO",1,3,0
2790 DATA"BALEARES:[SPC]AUMENTA[SPC]
TURISMO",1,3,0
2800 DATA"MURCIA:[SPC]AUMENTA[SPC]TU
RISMO",1,3,0
2810 DATA"CANARIAS:[SPC]AUMENTA[SPC]
TURISMO",1,3,0
2820 DATA"ANDALUCIA:[SPC]AUMENTA[SPC]
TURISMO",1,3,0
2830 DATA"MURCIA:[SPC]TERREMOTO",-1,
-2,0
2840 DATA"GRANADA:[SPC]TERREMOTO",-1
,-2,0
2850 DATA"CANARIAS:[SPC]TERREMOTO",-
1,-2,0
2860 DATA"CANARIAS:[SPC]MAREMOTO",-1
,-2,0
2870 DATA"GALICIA","ASTURIAS","CANTA
BRIA","PAIS[SPC]VASCO","NAVARRA","AR
AGON"
2880 DATA"CATALUNA","CASTILLA[SPC]LE
ON","RIOJA","MADRID","CASTILLA[SPC]L
A[SPC]MANCHA"
2890 DATA "VALENCIA","BALEARES","EXT
REMADURA","ANDALUCIA","MURCIA","CANA
RIAS"
2900 DATA 8,11,9,16,12,15,4,2,2,3,7,
12,13,16,17,15,16,15,17,17
2910 FOR H1=54272TO 54296:POKE H1,0:
NEXT:POKE 54277,9:POKE54278,64:POKE5
4296,15
2920 PRINT"[CLR]":GOSUB 1950:FOR H1=
1TO 100:NEXT:GOSUB 2300
2930 HF=12 :LF=216:DR=2:VD=80:GOSUB
3670
2940 HF=11  :LF=114:DR=2:GOSUB 3670
2950 HF=10  :LF=205:DR=2:GOSUB 3670
2960 HF=11  :LF=114:DR=2:GOSUB 3670
2970 HF=12  :LF=216:DR=4:GOSUB 3670
2980 HF=14  :LF=107:DR=2:GOSUB 3670
2990 HF=16  :LF= 47:DR=2:GOSUB 3670
3000 HF=17  :LF= 37:DR=4:GOSUB 3670
3010 HF=19  :LF= 63:DR=4:GOSUB 3670
3020 HF=17  :LF= 37:DR=4:GOSUB 3670
3030 HF=16  :LF= 47:DR=4:GOSUB 3670
3040 HF=12  :LF=216:DR=2:GOSUB 3670
3050 HF=11  :LF=114:DR=2:GOSUB 3670
3060 HF=10  :LF=205:DR=2:GOSUB 3670
3070 HF=11  :LF=114:DR=2:GOSUB 3670
3080 HF=12  :LF=216:DR=4:GOSUB 3670
3090 HF=14  :LF=107:DR=2:GOSUB 3670
3100 HF=16  :LF= 47:DR=2:GOSUB 3670
3110 HF=17  :LF= 37:DR=4:GOSUB 3670
3120 HF=14  :LF=107:DR=4:GOSUB 3670
3130 HF=12  :LF=216:DR=4:GOSUB 3670
3140 HF=0   :LF=0  :DR=4:GOSUB 3670
```

```
3150 PRINT"[CLR]":GOSUB 1950:FOR H1=
1TO 200:NEXT:GOSUB 2330
3160 HF=10 :LF=205:DR=8:VD=50:GOSUB
3670
3170 HF=12 :LF=216:DR=4:GOSUB 3670
3180 HF=17 :LF= 37:DR=8:GOSUB 3670
3190 HF=16 :LF= 47:DR=4:GOSUB 3670
3200 HF=19 :LF= 63:DR=4:GOSUB 3670
3210 HF=17 :LF= 37:DR=20:GOSUB 3670
3220 HF=16 :LF= 47:DR=8:GOSUB 3670
3230 HF=14 :LF=107:DR=4:GOSUB 3670
3240 HF=16 :LF= 47:DR=4:GOSUB 3670
3245 HF=14 :LF=107:DR=4:GOSUB 3670
3250 HF=12 :LF=216:DR=4:GOSUB 3670
3260 HF=11 :LF=114:DR=12:GOSUB 3670
3270 PRINT"[CLR]":GOSUB 1950:FOR H1=
1TO 200:NEXT:GOSUB 2370
3280 VD=150:HF=0  :LF=0  :DR= 2:GOSU
B 3670
3290 HF=8  :LF=147:DR= 2:GOSUB 3670
3300 HF=11 :LF=114:DR= 2:GOSUB 3670
3305 HF=14 :LF=107:DR=2:GOSUB 3670
3306 HF=17 :LF= 37:DR=6:GOSUB 3670
3307 HF=14 :LF=107:DR=2:GOSUB 3670
3308 HF=11 :LF=114:DR= 2:GOSUB 3670
3310 HF=15 :LF= 70:DR=2:GOSUB 3670
3320 HF=14 :LF=107:DR=6:GOSUB 3670
3360 HF=12 :LF=216:DR=2:GOSUB 3670
3370 HF=8  :LF=147:DR= 2:GOSUB 3670
3380 HF=12 :LF=216:DR=2:GOSUB 3670
3390 HF=11 :LF=114:DR=12:GOSUB 3670
3400 PRINT"[CLR]":GOSUB 1950:FOR H1=
1TO 200:NEXT:GOSUB 2400
3410 VD= 70:HF=0  :LF=0  :DR= 2:GOSU
B 3670
3420 HF=12 :LF=216:DR= 2:GOSUB 3670
3430 HF=14 :LF=107:DR=2:GOSUB 3670
3440 HF=16 :LF=47  :DR= 2:GOSUB 3670
3450 HF=17 :LF=37  :DR= 2:GOSUB 3670
3460 HF=19 :LF= 63:DR=2:GOSUB 3670
3470 HF=17 :LF=37  :DR=6  :GOSUB 3670
3480 HF=19 :LF= 63:DR=1:GOSUB 3670
3490 HF=17 :LF=37  :DR=1  :GOSUB 3670
3500 HF=16 :LF=47  :DR= 2:GOSUB 3670
3510 HF=14 :LF=107:DR=1:GOSUB 3670
3520 HF=16 :LF=47  :DR= 1:GOSUB 3670
3530 HF=17 :LF=37  :DR= 2:GOSUB 3670
3540 HF=0  :LF=0   :DR= 2:GOSUB 3670
3550 HF=0  :LF=0   :DR= 2:GOSUB 3670
3560 HF=10 :LF=205:DR=2:GOSUB 3670
3570 HF=11 :LF=114:DR=2:GOSUB 3670
3580 HF=14 :LF=107:DR=2:GOSUB 3670
3590 HF=12 :LF=216:DR=6:GOSUB 3670
3600 HF=11 :LF=114:DR=1:GOSUB 3670
3610 HF=10 :LF=205:DR=1:GOSUB 3670
3620 HF=9  :LF=159:DR=2:GOSUB 3670
3630 HF=10 :LF=205:DR=1:GOSUB 3670
3640 HF=9  :LF=159:DR=1:GOSUB 3670
3650 HF=8  :LF=147:DR=12:GOSUB 3670
3660 FOR H1=54272TO 54296:POKE H1,0:
NEXT:RETURN
3670 POKE 54273,HF:POKE 54272,LF:POK
E 54276,17:FOR H1=1TO VD*DR:NEXT H1
3680 POKE 54276,16:RETURN
3690 FOR H1=54272TO54296:POKE H1,0:N
EXT:POKE 54277,13:POKE54278,33:POKE5
4296,7
3700 POKE 54275, 4  :POKE54274,255
3710 VD=60:HF=11 :LF=114:DR=2:GOSUB
3830
3720 HF=12 :LF=216:DR=2:GOSUB 3830
3730 HF=13 :LF=156:DR=14:GOSUB 3830
3740 HF=0  :LF=0   :DR=6 :GOSUB 3830
3750 HF=17 :LF=37  :DR=2 :GOSUB 3830
3760 HF=19 :LF= 63:DR=2:GOSUB 3830
3770 HF=20 :LF=100:DR=24:GOSUB 3830
3780 HF=0  :LF=0   :DR=3 :GOSUB 3830
3790 HF=13 :LF=156:DR=6:GOSUB 3830
3800 HF=15 :LF= 70:DR=3:GOSUB 3830
3810 HF=17 :LF=37 :DR=18:GOSUB 3830
3820 FOR H1=54272TO54296:POKE H1,0:N
EXT:RETURN
3830 POKE 54273,HF:POKE 54272,LF:POK
E 54276,65:FOR H1=1TO VD*DR:NEXT H1
3840 POKE 54276,64:RETURN
```

Cuatro en raya

Miguel Forgas Serra
Tiras, nº 14
Vilanova i la Geltrú (Barcelona)

VIC-20



Os envío el juego CUATRO EN RAYA. Su funcionamiento es muy sencillo, al inicio aparece en la pantalla la presentación del juego esperando que pulsemos una tecla cualquiera. Seguidamente aparecera el tablero, encima aparecerá parpadeando un cursor rojo que es el que indica la columna por donde caerá la ficha al pulsar la tecla return, este cursor se puede hacer desplazar a izquierda y derecha utilizando un par de teclas (que se encuentran una al lado de la otra, y que no sean teclas con funciones especiales). Este juego también funciona utilizando los paddles, uno para cada jugador.

Siempre empieza el juego el jugador que tenga el color rojo, y después el del azul y así sucesivamente hasta que uno de los dos jugadores consiga hacer el cuatro en raya, que entonces el VIC avisará de que el jugador que ha tirado en último lugar ha hecho el cuatro en raya; y preguntará si se desea hacer otra partida o no.

A continuación paso a explicaros lo que realiza cada grupo de líneas del listado:

Líneas de la 10 a la 80 son las que crean los caracteres.

Líneas de la 90 a la 180 son las que preparan al VIC para iniciar el programa.

Líneas de la 200 a la 320 son las que controlan a qué jugador le toca tirar.

Líneas de la 1000 a la 1200 es la subrutina que lee el teclado y los paddles.

Líneas de la 2000 a la 2510 es la subrutina que hace caer la ficha.

Líneas de la 3000 a la 3510 es la subrutina que comprueba si se ha hecho el cuatro en raya.

Líneas de la 9000 a la 9999 es la subrutina que indica que se ha hecho el cuatro en raya y pregunta si se quiere volver a jugar.

Si tu televisor es en blanco y negro es difícil de distinguir las

fichas de cada jugador, para solucionar este problema se debe cambiar los valores designados a la variable "I" en las siguientes líneas:
En la 200 y la 300 se debe asignar a la variable "I" el valor del color que se desee utilizar.
En las líneas 310, 320 y 1050 se debe comparar la variable "I" con el primer valor. Y en las líneas 315, 325 y 1660 se debe igualar la variable "I" con el 2º valor.
Este programa no necesita ampliación.

```
10 DATA 27,85,85,86,86,90,90,90,90,9
0,90,90,90,86,86,85,85
20 DATA 28,85,85,149,149,165,165,165
,165,165,165,165,165,149,149,85,85
30 DATA 0,255,255,255,255,255,255,25
5,255,255,255,255,255,255,255,255,25
5
40 DATA 29,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0
45 DATA -24,1,24,0,-22,1,22,0,-20,1,
20,0,-2,1,2,0,-1,-1
50 A=7168:B=7679:FORC=ATOBSTEP2:P=PE
EK(32768+(C-A)/2)
60 POKEC,P:POKEC+1,P:NEXT
70 FORY=1TO4:READF
80 FORT=0TO15:READS:POKEA+F*16+T,S:N
EXTT,Y
90 POKE36879,27:POKE36864,0:PRINTCHR
$(8)
100 POKE36865,158:POKE36878,15:POKE3
6867,25:POKE36869,255:POKE52,28:POKE
56,28
110 PRINT"[BLK][CLR][3CRSRD]]]]]CUAT
RO[2CRSRD]EN[2CRSRD]RAYA"
120 PRINT"[HOM][2CRSRD][22COMME]"
130 PRINT"[HOM][8CRSRD][22COMME]"
140 FORT=0TO12:POKE36864,T:POKE36865
,158-T*10:POKE36876,128+T*10
145 FORY=1TO60:NEXT:NEXT:POKE36876,0
:POKE198,0:WAIT198,1
150 POKE37139,0:CLR:POKE65,239:POKE6
6,16
160 A=7702:D=38422:P=D-22:R1=36872:R
2=36873
170 FORT=0TO241STEP2:POKED+T,9:POKED
+T+1,9:POKEA+T,27:POKEA+T+1,28:NEXT
180 PRINT"[HOM][CYN]@@@@@@@@@@@@@@@@
@@@@@@"
200 I=2:GOSUB1000:Q=R*2
210 POKEP+Q,I:POKEP+Q+1,I:FORT=1TO10
0:NEXT:POKEP+Q,3:POKEP+Q+1,3
220 IF S THENGOSUB2000:GOTO240
230 GOTO310
240 IFS=9THEN310
250 GOSUB3000:GOTO320
300 I=6:GOSUB1000:Q=R*2:GOTO210
310 IFI=2THEN200
315 IFI=6THEN300
320 IFI=2THEN300
325 IFI=6THEN200
1000 M=PEEK(197):S=0
1005 IFM=64THEN1040
1010 IFM=15THENS=1:RETURN
1020 IFM<32ANDR>0THENR=R-1
1030 IFM>31ANDR<10THENR=R+1
1040 IFPEEK(R1)=255ANDPEEK(R2)=255TH
EN RETURN
1050 IFI=2THEN1100
1060 IFI=6THEN1200
1070 RETURN
1100 POKE37154,127:R=INT(PEEK(R2)/25
)
1110 S=-((PEEK(37152)AND128)=0):POKE
37154,255:RETURN
1200 R=INT(PEEK(R1)/25):S=-((PEEK(37
137)AND16)=0):RETURN
2000 O=D
2005 IF(PEEK(O+Q)AND15)<>9THENS=9:RE
TURN
2010 IF(PEEK(O+Q)AND15)=9THEN2500
2020 O=O-22:POKEO+Q,I+8:POKEO+Q+1,I+
8
2030 POKE36877,195:POKE36876,0:FORT=
1TO25:NEXT:POKE36877,0:RETURN
2500 POKEO+Q,I+8:POKEO+Q+1,I+8:POKE3
6876,135+I*10:FORT=1TO20:NEXT:POKEO+
Q,9:POKEO+Q+1,9
2510 O=O+22:GOTO2010
3000 READCO,SU:A1=CO:ER=R
3005 IFSU=-1THEN POKE65,239:POKE66,1
6:RETURN
3010 IFSUTHENSU=0:CU=0
3020 IF(PEEK(O+Q+CO)AND15)=8+ITHEN34
00
3040 GOTO3000
3400 ER=ER+1*(A1=-24)+1*(A1=-2)+1*(A
1=20)-1*(A1=-20)-1*(A1=2)-1*(A1=24)
3450 IFR<3ANDER<0THEN3000
3460 IFR>7ANDER>10THEN3000
3500 CU=CU+1:IFCU>2THENRESTORE:GOTO9
000
3510 CO=CO+A1:GOTO3020
9000 FORT=0TO7:POKE36876,130+T*15:FO
RY=1TO90:NEXT
9100 POKE36875,135:POKE36879,24+(T+1
*(T=1)):FORY=1TO60:NEXTY,T
9200 POKE36876,0:POKE36875,0:POKE368
79,27
9240 PRINT"[BLK][HOM][5CRSRD]]]]]OTR
A]PARTIDA]]]]][2CRSRD]]]]]]]]SI]O]N
O]]]]]]]]"
9250 B$="]]]CUATRO]EN]RAYA]]]":C$="
@@@@@@@@@@@@@@@@@@@@@@"
9300 B1=1:POKE198,0
9310 PRINT"[HOM][BLK]"MID$(B$,1,B1)"
[CYN]"MID$(C$,B1):POKE36874,253:FORT
=1TO30:NEXT:POKE36874,0
9315 IFB1=21THENB1=0
9320 GETA$:IFA$=""THEN B1=B1+1:GOTO9
310
9330 IFA$="S"THENPRINT"[CLR]":GOTO15
0
9340 IFA$<>"N"THEN 9320
9999 PRINT"[BLU][CLR]":POKE36867,46:
POKE36866,150:POKE36865,38:POKE36864
,12:POKE36869,240:END
```



Superexpander

El cartucho superexpander no siempre es compatible con los caracteres programables como los vic standards. Para solucionarlo hay que hacer:

POKE 51,0:POKE 52,30:POKE55,0:POKE56,30

Lo más importante, no afecta a la memoria extra.

R.M.B. RUN EE.UU.

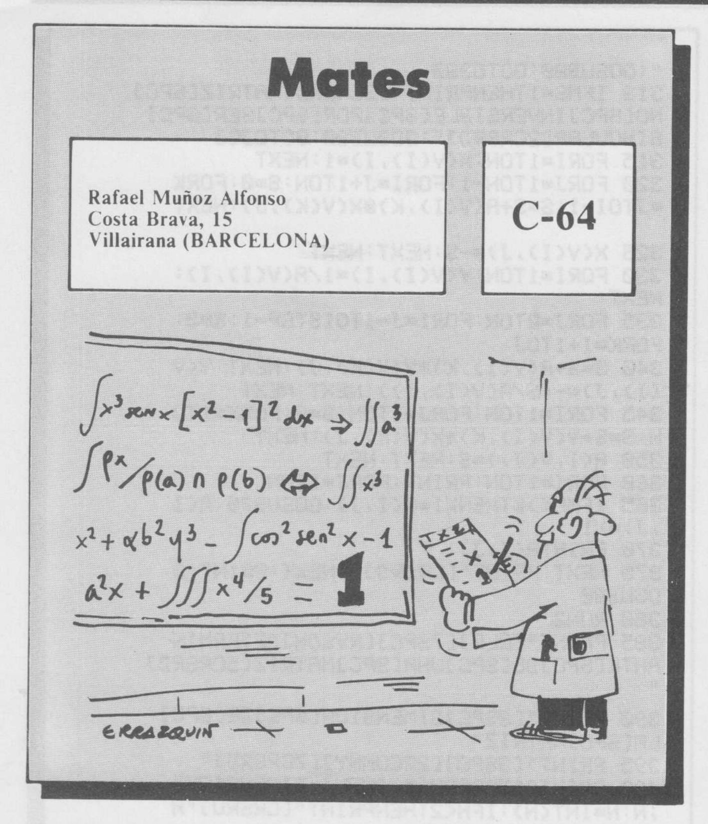
Bucles de retardo

Para colocar bucles de retardo, prefiero usar el reloj interno del ordenador, en la siguiente forma:

```
100 t=ti
110 ifti  t+120then110
```

En este caso, el retardo será de 2 segundos, que es igual a 120 sesentaavos de segundo.

E.H. RUN EE.UU.



Mates

Rafael Muñoz Alfonso
Costa Brava, 15
Villairana (BARCELONA)

C-64

Este programa, que consiste en un conjunto de ocho, todos ellos de carácter matemático. Por este motivo, he denominado a este paquete de programas con el nombre de "MATES".

El programa ha sido preparado con un C-64, aunque podría funcionar en un VIC-20 realizando los oportunos "recortes" por cuestiones de memoria y tamaño de la pantalla.

Este paquete de programas permite operar con dos matrices: sumándolas, restándolas o multiplicándolas. También permite calcular la matriz traspuesta de otra dada, así como su matriz inversa y su determinante. Además, con "MATES" se pueden resolver sistemas de ecuaciones lineales de coeficientes reales y ecuaciones polinómicas de coeficientes complejos.

El programa "MATES" ofrece la posibilidad de presentar los resultados con el número de decimales que deseemos, obteniendo de esta manera más claridad en los resultados finales.

He de indicar, que en la subrutina que triangulariza matrices hay dos condicionales (líneas 160 y 165), que utilizan una aproximación numérica (-1E-5<S>1E-5). Esto es así, debido a que, aunque la triangularización es un proceso exacto (no utiliza ni aproximaciones ni redondeos), con matrices de cierto tamaño (15×15 por ejemplo) en las que se efectúan muchas operaciones, existen errores de cálculo que se van acumulando durante la triangularización, pero estos errores son innatos del propio ordenador. Por este motivo, he utilizado este rango de error, ya que matrices de dimensión superior a 15 que deberían ser singulares, con el condicional S=0, no lo eran. Sin embargo, admito que el rango de error es quizás demasiado grande, pero esto se subsana sin más que variar el rango a otro menor, e incluso cambiar las comparaciones de las líneas mencionadas por S=0 si las matrices no son muy "grandes".

Al escribir el programa intenté que la entrada de datos fuese lo más clara posible, por lo que en todo momento en la pantalla se especifica el dato a introducir.

Debido a la extensión del programa, sería bastante laborioso explicarlo, por lo que paso a detallar, más o menos, qué operación realiza cada conjunto de líneas.

Línea 1	: Presentación del programa.
Líneas 2-9	: Se ofrece la posibilidad de redondeo de decimales.
Líneas 10-65	: Presentación del menú del programa.
Líneas 70-75	: Subrutina de redondeo de decimales.
Líneas 80-90	: Detención del programa para volver al menú.
Líneas 95-190	: Subrutina que triangulariza matrices.
Líneas 195-275	: Entrada de datos.
Líneas 280-380	: Cálculo de la inversa de una matriz.
Líneas 385-445	: Cálculo del determinante de una matriz.
Líneas 450-535	: Resolución de sistemas lineales de ecuaciones.
Líneas 540-665	: Suma y resta de matrices.
Líneas 670-795	: Producto de matrices.
Líneas 800-865	: Traspuesta de una matriz.
Líneas 866-990	: Resolución de ecuaciones polinómicas complejas.
Líneas 1000-1300	: Presentación del programa.

```
1 GOSUB1000
2 PRINT"[CLR][6CRSRD][4SPC]QUIERES[SPC]
REDONDEAR[SPC]DECIMALES[SPC][RVSON](
S/N)[RVSOFF]?"
3 GETG$:IFG$<>"S"ANDG$<>"N"THEN3
4 H$="[34SPC]":IFG$="N"THENMM=0:GOTO
10
5 PRINT"[CLR][4SPC][RVSON]REDONDEO[SPC]
DE[SPC]DECIMALES[SPC]HASTA[SPC]LAS[3CRSRD]
":PRINTTAB(12)"[RVSON]1[RVSOFF]-DECI
MAS[CRSRD]"
6 PRINTTAB(12)"[RVSON]2[RVSOFF]-CENT
ESIMAS[CRSRD]":PRINTTAB(12)"[RVSON]3
[RVSOFF]-MILESIMAS[CRSRD]"
7 PRINTTAB(12)"[RVSON]4[RVSOFF]-DIEZ
MILESIMAS[CRSRD]":PRINTTAB(12)"[RVSON]
5[RVSOFF]-CIENMILESIMAS[5CRSRD]"
8 INPUT"REDONDEO";I:I=INT(I):IFI<1OR
I>5THENPRINT"[CRSRU]"H$"[CRSRU]":GOT
O8
9 "OMNIBUS[6SPC]ROM" PRG
10 PRINT"[CLR]"TAB(18)"[RVSON]MENU[CRSRD]
"
15 PRINTTAB(5)"[RVSON]1[RVSOFF]-SUMA
[SPC]DE[SPC]MATRICES[SPC][RVSON][A]+
[B][CRSRD]"
20 PRINTTAB(5)"[RVSON]2[RVSOFF]-REST
A[SPC]DE[SPC]MATRICES[SPC][RVSON][A]
-[B][CRSRD]"
25 PRINTTAB(5)"[RVSON]3[RVSOFF]-PROD
UCTO[SPC]DE[SPC]MATRICES[SPC][RVSON]
[A]*[B][CRSRD]"
30 PRINTTAB(5)"[RVSON]4[RVSOFF]-TRAS
PUESTA[SPC]DE[SPC]UNA[SPC]MATRIZ[CRSRD]
"
35 PRINTTAB(5)"[RVSON]5[RVSOFF]-DETE
RMINANTE[SPC]DE[SPC]UNA[SPC]MATRIZ[CRSRD]
"
40 PRINTTAB(5)"[RVSON]6[RVSOFF]-INVE
RSA[SPC]DE[SPC]UNA[SPC]MATRIZ[CRSRD]
"
45 PRINTTAB(5)"[RVSON]7[RVSOFF]-SIST
EMAS[SPC]LINEALES[SPC]DE[SPC]ECUACIO
NES[CRSRD]"
50 PRINTTAB(5)"[RVSON]8[RVSOFF]-ECUA
CIONES[SPC]POLINOMICAS[CRSRD]"
52 PRINTTAB(5)"[RVSON]9[RVSOFF]-...
.[RVSON]FIN[SPC]DE[SPC]TRABAJO[RVSOFF]
.....[2CRSRD]"
55 PRINT"PULSA[SPC]EL[SPC]CODIGO[SPC]
QUE[SPC]DESEES[CRSRD]"
60 INPUT"[RVSON]CODIGO[RVSOFF]";CO:C
O=INT(CO):IFCO<1ORCO>9THENPRINT"[CRSRU]
"H$"[CRSRU]":GOTO60
65 ONCOGOTO540,590,670,800,385,280,4
50,866,1300
70 X1=INT(X1*MM+.5)/MM
75 RETURN
80 PRINT"[CRSRD][RVSON]PARA[SPC]VOLV
ER[SPC]AL[SPC]MENU[SPC]PULSAR[RVSOFF]
[SPC]M"
85 GETAA$:IFAA$<>"M"THEN85
90 RETURN
95 FORI=1TON:V(I)=I:NEXT
100 I=1:Q=1
```

```
105 IFI=NANDA(V(Q),1)=0THENMN=1:GOTO
185
110 IFA(V(Q),1)=0THENL1=L1+1:C=V(Q):
V(Q)=V(Q+I):V(Q+I)=C:I=I+1:GOTO105
115 FORI=2TON:Q=0
120 FORJ=1TON:Z(J)=A(V(I),J):NEXT
125 FORJ=1TON
130 IFI-1=<J-1THENM=I-1
135 IFI-1>J-1THENM=J-1
140 IFM=0THEN175
145 S=0:FORK=1TOM:S=S+Z(K)*A(V(K),J)
:NEXT:S=Z(J)-S
150 IFJ<ITHENS=S/A(V(J),J)
155 Z(J)=S
160 IFI=JAND(-1E-5<SANDS<1E-5)ANDI+Q
+1>NTHENMS=1:GOTO185
165 IFI=JAND(-1E-5<SANDS<1E-5)THENL1
=L1+1:Q=Q+1:GOSUB190:GOTO120
170 GOTO180
175 Z(J)=Z(J)/A(V(J),J)
180 NEXT:FORJ=1TON:A(V(I),J)=Z(J):NE
XT:NEXT
185 RETURN
190 C=V(I):V(I)=V(I+Q):V(I+Q)=C:RETU
RN
195 INPUT"NUMERO[SPC]DE[SPC]FILAS";N
:N=INT(N)
200 IFN<2THENPRINT"[CRSRU]"H$"[CRSRU]
":GOTO195
205 PRINT
210 INPUT"NUMERO[SPC]DE[SPC]COLUMNAS
";M1:M1=INT(M1)
215 IFM1<2THENPRINT"[CRSRU]"H$"[CRSRU]
":GOTO210
220 RETURN
225 FORI=1TON:PRINT"[CRSRD]FILA"I"[SPC]
DE[SPC][RVSON][A][RVSOFF][2CRSRD]":F
ORJ=1TOM1:PRINT"ELEMENTO[SPC]A("I"[SPC]
,"J"[SPC])"
230 INPUTA(I,J):PRINT"[2CRSRU]"H$:PR
INTH$"[2CRSRU]":NEXT
235 PRINT"[5CRSRU]":NEXT
240 RETURN
245 FORI=1TON:PRINT"[CRSRD]FILA"I"[SPC]
DE[SPC][RVSON][A][RVSOFF][2CRSRD]":F
ORJ=1TOM1:PRINT"ELEMENTO[SPC]A("I"[SPC]
,"J"[SPC])"
250 INPUTA(I,J):PRINT"[2CRSRU]"H$:PR
INTH$"[2CRSRU]":NEXT
255 PRINT"[5CRSRU]":NEXT:PRINT"[CRSRD]
"H$"[2CRSRU]"
260 FORI=1TOM2:PRINT"[CRSRD]FILA"I"[SPC]
DE[SPC][RVSON][B][RVSOFF][2CRSRD]":F
ORJ=1TOM3:PRINT"ELEMENTO[SPC]B("I"[SPC]
,"J"[SPC])"
265 INPUTX(I,J):PRINT"[2CRSRU]"H$:PR
INTH$"[2CRSRU]":NEXT
270 PRINT"[5CRSRU]":NEXT
275 RETURN
280 PRINT"[CLR][10SPC][RVSON]INVERSA
[SPC]DE[SPC]UNA[SPC]MATRIZ[4CRSRD]":
PRINT"[SPC]DIMENSION[SPC]DE[SPC]LA[SPC]
MATRIZ"
285 INPUTN:N=INT(N):IFN<2THENPRINT"[CRSRU]
"H$"[CRSRU]":GOTO285
290 PRINT"[4CRSRD]":DIMA(N,N),Y(N,N)
,Z(N),V(N)
295 FORI=1TON:PRINT"FILA"I"[5CRSRD]"
:FORJ=1TON:PRINT"[2CRSRU]ELEMENTO[SPC]
A("I"[SPC],"J"[SPC])"
300 INPUTA(I,J):PRINT"[CRSRU]"H$:NEX
T:PRINT"[2CRSRU]"H$
305 PRINT"[6CRSRU]":NEXT
307 PRINT"[CLR][10SPC][RVSON]INVERSA
[SPC]DE[SPC]UNA[SPC]MATRIZ[CRSRD]":P
RINT"[SPC]RESULTADO[CRSRD]"
310 MN=0:MS=0:GOSUB95
312 IFMN=1THENPRINT"[5CRSRD]LA[SPC]M
ATRIZ[SPC]NO[SPC]ES[SPC]CUADRADA[3CRSRD]
":GOSUB80:GOTO380
313 IFMS=1THENPRINT"[3CRSRD]MATRIZ[SPC]
NO[SPC]INVERSIBLE[SPC]POR[SPC]SER[SPC]
SINGULAR[2CRSRD]":GOSUB80:GOTO380
315 FORI=1TON:X(V(I),I)=1:NEXT
320 FORJ=1TON-1:FORI=J+1TON:S=0:FORK
=JTOI-1:S=S+A(V(I),K)*X(V(K),J):NEXT
325 X(V(I),J)=-S:NEXT:NEXT
330 FORI=1TON:Y(V(I),I)=1/A(V(I),I):
NEXT
335 FORJ=2TON:FORI=J-1TO1STEP-1:S=0:
FORK=I+1TOJ
340 S=S+A(V(I),K)*Y(V(K),J):NEXT:Y(V
(I),J)=-(S/A(V(I),I)):NEXT:NEXT
345 FORI=1TON:FORJ=1TON:S=0:FORK=1TO
N:S=S+Y(V(I),K)*X(V(K),J):NEXT
350 A(I,V(J))=S:NEXT:NEXT
360 FORI=1TON:PRINT:FORJ=1TON
365 IFMM<>0THENX1=A(I,J):GOSUB70:A(I
,J)=X1
370 PRINTA(I,J);
375 NEXT:PRINT"[CRSRD]":NEXT:PRINT:G
OSUB80
380 RUN2
385 PRINT"[CLR][7SPC][RVSON]DETERMIN
ANTE[SPC]DE[SPC]UNA[SPC]MATRIZ[5CRSRD]
"
390 PRINT"[3SPC]DIMENSION[SPC]DE[SPC]
LA[SPC]MATRIZ"
395 PRINT"[3SPC][22COMMY][7CRSRD]"
400 PRINT"[5CRSRU]":INPUT"DIMENSION"
;N:N=INT(N):IFN<2THENPRINT"[CRSRU]"H
$"[3CRSRD]":GOTO400
410 DIMA(N,N),V(N),Z(N):M1=N:PRINT"[5CRSRD]
":GOSUB225:MN=0:MS=0
415 L1=0:PRINT"[CLR][7SPC][RVSON]DET
ERMINANTE[SPC]DE[SPC]UNA[SPC]MATRIZ[7CRSRD]
"
420 PRINT"RESULTADO[CRSRD]":GOSUB95
422 IFMN=1THENPRINT"[2CRSRD]LA[SPC]M
ATRIZ[SPC]NO[SPC]ES[SPC]CUADRADA[3CRSRD]
":GOTO440
423 IFMS=1THENPRINT"[3CRSRD][3SPC]DE
TERMINANTE[SPC]=[SPC]0[4CRSRD]":GOTO
440
425 DET=1:FORI=1TON:DET=DET*A(V(I),I
):NEXT:DET=DET*(-1)↑L1
430 IFMM<>0THENX1=DET:GOSUB70:DET=X1
435 PRINT"[3CRSRD][3SPC]DETERMINANTE
[SPC]=";DET:PRINT"[3CRSRD]"
440 GOSUB80
445 RUN2
450 PRINT"[CLR][4SPC][RVSON]SISTEMAS
[SPC]LINEALES[SPC]DE[SPC]ECUACIONES"
:PRINTTAB(15)"[RVSON][A]*[X]=[Y][CRSRD]
"
451 PRINT"[RVSON][A][RVSOFF]:MATRIZ[SPC]
DE[SPC]COEFICIENTES[CRSRD]":PRINT"[RVSON]
[X][RVSOFF]:VECTOR[SPC]DE[SPC]INCOGN
ITAS[CRSRD]"
452 PRINT"[RVSON][Y][RVSOFF]:VECTOR[SPC]
DE[SPC]TERMINOS[SPC]INDEPENDIENTES[3CRSRD]
"
453 PRINT"DIMENSION[SPC]DEL[SPC]SIST
EMA?[SPC](O[SPC]DE[SPC][RVSON][A][RVSOFF]
)"
455 INPUTN:N=INT(N):IFN<2THENPRINT"[CRSRU]
"H$"[CRSRU]":GOTO455
460 PRINT"[3CRSRD]":DIMA(N,N),Y(N),Z
(N),V(N),X(N)
465 FORI=1TON:PRINT"FILA"I"[5CRSRD]"
:FORJ=1TON:PRINT"[2CRSRU]ELEMENTO[SPC]
A("I"[SPC],"J"[SPC])"
470 INPUTA(I,J):PRINT"[CRSRU]"H$:NEX
T:PRINT"[2CRSRU]"H$
475 PRINT"[CRSRU]TERMINO[SPC]Y("I")"
:INPUTY(I):PRINT"[2CRSRU]"H$:PRINTH$
```

```
:PRINT"[7CRSRU]":NEXT
477 PRINT"[CLR][4SPC][RVSON]SISTEMAS
[SPC]LINEALES[SPC]DE[SPC]ECUACIONES[CRSRD]
"
478 PRINT"SOLUCIONES[SPC]DEL[SPC]SIS
TEMA[CRSRD]"
480 MN=0:MS=0:GOSUB95
483 IFMN=1ORMS=1THENPRINT"[4CRSRD][10CRSRR]
SISTEMA[SPC]INDETERMINADO[4CRSRD]":G
OTO533
485 Z(V(1))=Y(V(1))
490 FORI=2TON:Z(V(I))=Y(V(I)):FORK=1
TOI-1:Z(V(I))=Z(V(I))-A(V(I),K)*Z(V(
K))
495 NEXT:NEXT
500 X(V(N))=Z(V(N))/A(V(N),N)
505 FORI=N-1TO1STEP-1:X(V(I))=Z(V(I)
)
510 FORK=I+1TON:X(V(I))=X(V(I))-A(V(
I),K)*X(V(K)):NEXT
512 X(V(I))=X(V(I))/A(V(I),I):NEXT
520 FORI=1TON:IFMM<>0THENX1=X(V(I)):
GOSUB70:X(V(I))=X1
522 IFI<>10THEN525
523 PRINT"[RVSON]PARA[SPC]CONTINUAR[SPC]
PULSAR[RVSOFF][SPC]I[CRSRD]"
524 GETEE$:IFEE$<>"I"THEN524
525 PRINTTAB(10)"X("I")=";X(V(I)):PR
INT
530 NEXT:PRINT
533 GOSUB80
535 RUN2
540 PRINT"[CLR][8SPC][RVSON]SUMA[SPC]
DE[SPC]MATRICES[SPC][A]+[B][2CRSRD]"

545 PRINT"[3SPC]DIMENSION[SPC]DE[SPC]
LAS[SPC]MATRICES"
550 PRINT"[3SPC][25COMMY][7CRSRD]":G
OSUB195
555 PRINT"[CLR][8SPC][RVSON]SUMA[SPC]
DE[SPC]MATRICES[SPC][A]+[B][5CRSRD]"

560 DIMA(N,M1),X(N,M1):M2=N:M3=M1
565 GOSUB245
570 PRINT"[CLR][8SPC][RVSON]SUMA[SPC]
DE[SPC]MATRICES[SPC][A]+[B][CRSRD]":
PRINT"RESULTADO[CRSRD]"
575 L=1:GOSUB645
580 GOSUB80
585 RUN2
590 PRINT"[CLR][8SPC][RVSON]RESTA[SPC]
DE[SPC]MATRICES[SPC][A]-[B][2CRSRD]"

595 PRINT"[3SPC]DIMENSION[SPC]DE[SPC]
LAS[SPC]MATRICES"
600 PRINT"[3SPC][25COMMY][2CRSRD]"
605 GOSUB195
610 PRINT"[CLR][8SPC][RVSON]RESTA[SPC]
DE[SPC]MATRICES[SPC][A]-[B][5CRSRD]"

615 DIMA(N,M1),X(N,M1):M2=N:M3=M1
620 GOSUB245
625 PRINT"[CLR][8SPC][RVSON]RESTA[SPC]
DE[SPC]MATRICES[SPC][A]-[B][CRSRD]":
PRINT"RESULTADO[CRSRD]"
630 L=-1:GOSUB645
635 GOSUB80
640 RUN2
645 FORI=1TON:FORJ=1TOM1:A(I,J)=A(I,
J)+L*X(I,J)
650 IFMM<>0THENX1=A(I,J):GOSUB70:A(I
,J)=X1
655 PRINTA(I,J);
660 NEXT:PRINT"[CRSRD]":NEXT
665 RETURN
670 PRINT"[CLR][6SPC][RVSON]PRODUCTO
[SPC]DE[SPC]MATRICES[SPC][A]*[B][2CRSRD]
"
675 PRINT"[3SPC]DIMENSION[SPC]DE[SPC]
LA[SPC]MATRIZ[SPC][RVSON][A]"
680 PRINT"[3SPC][26COMMY][2CRSRD]"
685 GOSUB195
690 PRINT"[2CRSRD][3SPC]DIMENSION[SPC]
DE[SPC]LA[SPC]MATRIZ[SPC][RVSON][B]"

695 PRINT"[3SPC][26COMMY][2CRSRD]"
700 INPUT"NUMERO[SPC]DE[SPC]FILAS";M
2:M2=INT(M2):IFM2<2THENPRINT"[CRSRU]
"H$"[CRSRU]":GOTO700
710 PRINT
715 INPUT"NUMERO[SPC]DE[SPC]COLUMNAS
";M3:M3=INT(M3):IFM3<2THENPRINT"[CRSRU]
"H$"[CRSRU]":GOTO715
725 IFM1=M2THEN750
730 PRINT"[CRSRD][RVSON]ERROR[RVSOFF]
[SPC]NO[SPC]SE[SPC]PUEDE[SPC]MULTIPL
ICAR[SPC]LA[SPC]MATRIZ"
735 PRINT"[RVSON]A[RVSOFF]("N","M1")
[SPC]POR[SPC]LA[2SPC][RVSON]B[RVSOFF]
("M2","M3")YA[SPC]QUE[SPC]LAS"
740 PRINT"COLUMNAS[SPC]DE[SPC][RVSON]
A[RVSOFF][SPC]NO[SPC]COINCIDEN[SPC]C
ON[SPC]LAS[SPC]FILAS[SPC]DE[SPC][RVSON]
B[RVSOFF]."
745 FORI=1TO9000:NEXT:GOTO670
750 DIMA(N,M1),X(M2,M3),Y(N,M3)
755 PRINT"[CLR][6SPC][RVSON]PRODUCTO
[SPC]DE[SPC]MATRICES[SPC][A]*[B][5CRSRD]
"
760 GOSUB245
765 PRINT"[CLR][6SPC][RVSON]PRODUCTO
[SPC]DE[SPC]MATRICES[SPC][A]*[B][CRSRD]
":PRINT"RESULTADO[CRSRD]"
770 FORI=1TON:FORJ=1TOM3:S=0:FORK=1T
OM1:S=S+A(I,K)*X(K,J):NEXT
775 Y(I,J)=S:IFMM<>0THENX1=Y(I,J):GO
SUB70:Y(I,J)=X1
780 PRINTY(I,J);
785 NEXT:PRINT"[CRSRD]":NEXT
790 GOSUB80
795 RUN2
800 PRINT"[CLR][7SPC][RVSON]TRASPUES
TA[SPC]DE[SPC]UNA[SPC]MATRIZ[2CRSRD]
"
805 PRINT"[3SPC]DIMENSION[SPC]DE[SPC]
LA[SPC]MATRIZ"
810 PRINT"[3SPC][22COMMY][7CRSRD]"
815 GOSUB195
820 PRINT"[CLR][7SPC][RVSON]TRASPUES
TA[SPC]DE[SPC]UNA[SPC]MATRIZ[5CRSRD]
"
825 DIMA(N,M1),X(M1,N)
830 GOSUB225
835 PRINT"[CLR][7SPC][RVSON]TRASPUES
TA[SPC]DE[SPC]UNA[SPC]MATRIZ[CRSRD]"
:PRINT"RESULTADO[CRSRD]"
840 FORJ=1TOM1:FORI=1TON:X(J,I)=A(I,
J)
845 IFMM<>0THENX1=X(J,I):GOSUB70:X(J
,I)=X1
850 PRINTX(J,I);
855 NEXT:PRINT"[CRSRD]":NEXT
860 GOSUB80
865 RUN2
866 PRINT"[CLR]"TAB(5)"[RVSON]ECUACI
ONES[SPC]POLINOMICAS[SPC]REALES":PRI
NTTAB(15)"[RVSON]Y[SPC]COMPLEJAS[3CRSRD]
"
867 INPUT"GRADO[SPC]POLINOMIO";N:N=I
NT(N):IFN<1THENPRINT"[CRSRU]"H$"[CRSRU]
":GOTO867
868 N1=N+1:N2=N:DIMAR(N1),AI(N1),BR(
N1),BI(N1),CR(N1),CI(N1)
869 DIMXR(N2),YR(N2),MO(N2),AG(N2):P
RINT"[2CRSRD]"
870 FORI=1TON1:PRINT"COEFICIENTE[SPC]
DE[SPC][RVSON]X↑"N1-I"[RVSOFF][CRSRL]
[14SPC]":PRINT
```

```
872 INPUT"[RVSON]PARTE[SPC]REAL[RVSOFF]
";AR(I):PRINT"[2CRSRD]":INPUT"[RVSON]
PARTE[SPC]IMAGINARIA[RVSOFF]";AI(I)
873 PRINT"[5CRSRU]"H$:PRINT"[3CRSRD]
"H$:PRINT"[8CRSRU]":NEXT:PRINTH$"[CRSRU]
"
875 INPUT"[RVSON]ERROR[SPC]ADMISIBLE
[RVSOFF]";EP:PRINT"[CLR]"TAB(5)"[RVSON]
ECUACIONES[SPC]POLINOMICAS[SPC]REALE
S"
877 PRINTTAB(15)"[RVSON]Y[SPC]COMPLE
JAS"
878 PRINT"[CRSRD]"TAB(5)"[RVSON]PART
E[SPC]REAL"TAB(22)"[RVSON]PARTE[SPC]
COMPLEJA[CRSRD]"
880 BR(1)=AR(1):BI(1)=AI(1):CR(1)=AR
(1):CI(1)=AI(1)
883 FORNR=1TON
886 X=0.:Y=1.
889 FORI=2TON1
892 BR(I)=AR(I)+BR(I-1)*X-BI(I-1)*Y
895 BI(I)=AI(I)+BR(I-1)*Y+BI(I-1)*X:
NEXT
898 FORI=2TON
901 CR(I)=BR(I)+CR(I-1)*X-CI(I-1)*Y
904 CI(I)=BI(I)+CR(I-1)*Y+CI(I-1)*X:
NEXT
907 DE=(CR(N)↑2+CI(N)↑2)+1E-25
910 X=X-(BR(N1)*CR(N)+BI(N1)*CI(N))/
DE
913 Y=Y+(BR(N1)*CI(N)-BI(N1)*CR(N))/
DE
916 IF(BR(N1)↑2+BI(N1)↑2-EP)>0THEN88
9
919 N1=N1-1:N=N-1
922 FORI=2TON1
925 AR(I)=BR(I):AI(I)=BI(I):NEXT
928 RO=SQR(X*X+Y*Y):FI=180.*ATN(Y/(A
BS(X)+1E-25))/π
931 IFX>=0THEN946
934 IFY>=0THEN943
937 FI=-180.-FI
940 GOTO946
943 FI=180.-FI
946 MO(NR)=RO:AG(NR)=FI:IFMM=0THEN94
8
947 X1=X:GOSUB70:X=X1:X1=Y:GOSUB70:Y
=X1
948 IFNR<>10THEN958
949 PRINT"[RVSON]PARA[SPC]CONTINUAR[SPC]
PULSAR[RVSOFF][SPC]I"
950 GETEE$:IFEE$<>"I"THEN950
951 PRINT
958 PRINTTAB(10-LEN(STR$(X))/2)XTAB(
30-LEN(STR$(Y))/2)Y"[CRSRD]":NEXT
960 PRINT"[RVSON]PULSAR[RVSOFF][SPC]
P[SPC][RVSON]PARA[SPC]LAS[SPC]SOLUCI
ONES[SPC]EN[SPC]POLARES"
962 GETP$:IFP$<>"P"THEN962
964 PRINT"[CLR]"TAB(5)"[RVSON]ECUACI
ONES[SPC]POLINOMICAS[SPC]REALES":PRI
NTTAB(15)"[RVSON]Y[SPC]COMPLEJAS"
966 PRINT"[CRSRD]"TAB(7)"[RVSON]MODU
LO"TAB(25)"[RVSON]ARGUMENTO[CRSRD]"
968 FORI=1TON2
970 IFMM=0THEN974
972 X1=MO(I):GOSUB70:MO(I)=X1:X1=AG(
I):GOSUB70:AG(I)=X1
974 IFI<>10THEN980
975 PRINT"[RVSON]PARA[SPC]CONTINUAR[SPC]
PULSAR[RVSOFF][SPC]I"
976 GETEE$:IFEE$<>"I"THEN976
977 PRINT
980 PRINTTAB(10-LEN(STR$(MO(I)))/2)M
O(I)TAB(30-LEN(STR$(AG(I)))/2)AG(I)"
[CRSRD]"
982 NEXT:PRINT"[2CRSRU]"
988 GOSUB80
990 RUN2
```

```
1000 PRINT"[CLR][8CRSRD]"TAB(9)"[RVSON]
PAQUETE[SPC]DE[SPC]PROGRAMAS[SPC]DE[CRSRD]
"
1002 PRINTTAB(11)"[RVSON]CARACTER[SPC]
MATEMATICO[9CRSRD]"
1004 PRINTTAB(20)"POR[SPC]RAFAEL[SPC]
MUNYOZ":FORI=1TO5000:NEXT
1006 PRINT"[CLR][8CRSRD]"TAB(5)"[RVSON]
DESEAS[SPC]ALGUNAS[SPC]INDICACIONES[SPC]
(S/N)?[CRSRD]"
1008 GETW$:IFW$<>"S"ANDW$<>"N"THEN10
08
1010 IFW$="N"THEN1136
1014 PRINT"[CLR][2SPC]ESTE[SPC]PAQUE
TE[SPC]DE[SPC]PROGRAMAS[SPC]TE[SPC]P
ERMITIRA[SPC]OPERAR[SPC]CON[SPC]DOS[SPC]
MATRICES,";
1016 PRINT"[SPC]SUMANDOLAS,[SPC]REST
ANDOLAS[SPC]O[SPC]MULTIPLICANDOLAS.[SPC]
SI[SPC]SE[SPC]REALIZA[SPC]ALGUNA[SPC]
";
1018 PRINT"DE[SPC]LAS[SPC]OPERACIONE
S[SPC]ANTERIORES,[SPC]EL[SPC]PROGRAM
A[SPC]EMPIEZA[SPC]PREGUNTANDO[SPC]LA
[SPC]";
1020 PRINT"DIMENSION[SPC]DE[SPC]LAS[SPC]
MATRICES,[SPC]O[SPC]SEA,[SPC]EL[SPC]
NUMERO[SPC]DE[SPC]FILAS[SPC]Y[SPC]CO
LUMNAS[SPC]";
1022 PRINT"DE[SPC]ELLAS.[SPC]NO[SPC]
SE[SPC]ADMITIRAN[SPC]MATRICES[SPC]UN
IDIMENSIONALES[SPC](VECTORES).[SPC]"
;
1024 PRINT"A[SPC]CONTINUACION,[SPC]S
E[SPC]INTRODUCIRAN[SPC]LOS[SPC]ELEME
NTOS[SPC]DE[SPC]CADA[SPC]MATRIZ[SPC]
POR[SPC]";
1026 PRINT"FILAS.[SPC]EN[SPC]TODO[SPC]
MOMENTO,[SPC]EN[SPC]LA[SPC]PANTALLA[SPC]
SE[SPC]ESPECIFICA[SPC]EL[SPC]ELEMENT
O[SPC]A";
1028 PRINT"[SPC]INTRODUCIR,[SPC]NO[SPC]
HABIENDO[SPC]ASI[SPC]POSIBILIDAD[SPC]
DE[SPC]ERROR[SPC]AL[SPC]INTRODUCIR[2SPC]
";
1030 PRINT"LOS[SPC]DATOS."
1032 PRINT"[2SPC]ADEMAS,[SPC]PODRAS[SPC]
OPERAR[SPC]CON[SPC]UNA[SPC]SOLA[SPC]
MATRIZ[SPC]CALCULANDO[SPC]SU[SPC]";
1034 PRINT"TRASPUSESTA,[SPC]SU[SPC]D
ETERMINANTE[SPC]O[SPC]SU[SPC]INVERSA
.[SPC]EN[SPC]ESTOS[SPC]CASOS,[SPC]LO
S[SPC]";
1036 PRINT"DATOS[SPC]SE[SPC]INTRODUC
EN[SPC]DE[SPC]LA[SPC]MISMA[SPC]FORMA
[SPC]ANTERIOR."
1038 PRINT"[4CRSRD][RVSON]PARA[SPC]C
ONTINUAR[SPC]PULSAR[RVSOFF][SPC]I"
1040 GETW$:IFW$<>"I"THEN1040
1042 PRINT"[CLR][4SPC]MEDIANTE[SPC]E
STE[SPC]PAQUETE[SPC]DE[SPC]PROGRAMAS
[SPC]PODRAS[SPC]RESOLVER[SPC]SISTEMA
S[SPC]";
1044 PRINT"LINEALES[SPC]DE[SPC]ECUAC
IONES[SPC]CON[SPC]COEFICIENTES[SPC]R
EALES.[SPC]TAL[SPC]Y[SPC]COMO[SPC]SE
[SPC]";
1046 PRINT"INDICA[SPC]EN[SPC]EL[SPC]
EJEMPLO[SPC]SIGUIENTE."
1048 PRINT"[CRSRD]"TAB(1)"[SHIFTO]"T
AB(26)"[SHIFTP]"TAB(28)"[SHIFTO][2SPC]
[SHIFTP][3SPC][SHIFTO][2SPC][SHIFTP]
"
1050 PRINTTAB(1)"[COMMJ]"TAB(2)"2.65
"TAB(7)"-.32"TAB(12)"9.48"TAB(17)"6.
43";
1052 PRINTTAB(22)".78"TAB(26)"[COMML]
"TAB(28)"[COMMJ]X1[COMML][3SPC][COMMJ]
34[COMML]"
1053 PRINTTAB(1)"[COMMJ]"TAB(26)"[COMML]
```

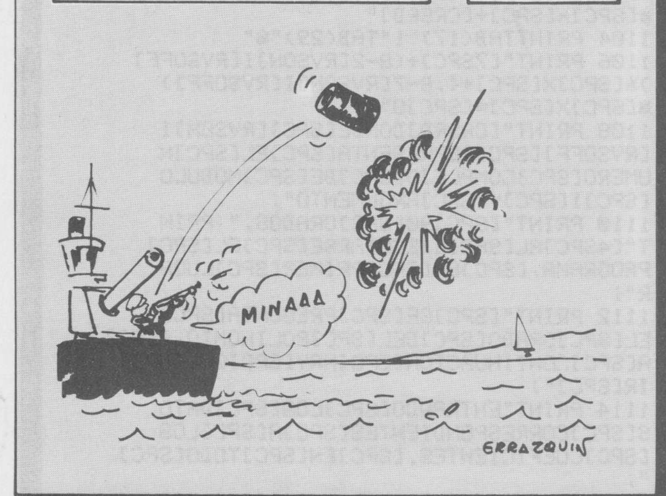
```
"TAB(28)"[COMMJ][2SPC][COMML][3SPC][COMMJ]
[2SPC][COMML]"
1054 PRINTTAB(1)"[COMMJ]"TAB(2)"-.33
"TAB(7)"5.83"TAB(12)"48.5"TAB(17)"21
.6";
1056 PRINTTAB(22)"4.25"TAB(26)"[COMML]
"TAB(28)"[COMMJ]X2[COMML][3SPC][COMMJ]
.9[COMML]"
1057 PRINTTAB(1)"[COMMJ]"TAB(26)"[COMML]
"TAB(28)"[COMMJ][2SPC][COMML][3SPC][COMMJ]
[2SPC][COMML]"
1058 PRINTTAB(1)"[COMMJ]"TAB(2)"89.5
"TAB(7)"7.43"TAB(12)"1.76"TAB(17)"-.
82";
1060 PRINTTAB(22)"55.8"TAB(26)"[COMML]
"TAB(28)"[COMMJ]X3[COMML][SPC]=[SPC]
[COMMJ]-7[COMML]"
1061 PRINTTAB(1)"[COMMJ]"TAB(26)"[COMML]
"TAB(28)"[COMMJ][2SPC][COMML][3SPC][COMMJ]
[2SPC][COMML]"
1062 PRINTTAB(1)"[COMMJ]"TAB(2)"-9.4
"TAB(7)"-8.2"TAB(12)"13.3"TAB(17)"-7
68";
1064 PRINTTAB(22)"-7.2"TAB(26)"[COMML]
"TAB(28)"[COMMJ]X4[COMML][3SPC][COMMJ]
83[COMML]"
1065 PRINTTAB(1)"[COMMJ]"TAB(26)"[COMML]
"TAB(28)"[COMMJ][2SPC][COMML][3SPC][COMMJ]
[2SPC][COMML]"
1066 PRINTTAB(1)"[COMMJ]"TAB(2)"3.92
"TAB(7)"87.4"TAB(12)"-421"TAB(17)"45
.6";
1068 PRINTTAB(22)"39.7"TAB(26)"[COMML]
"TAB(28)"[COMMJ]X5[COMML][3SPC][COMMJ]
.2[COMML]"
1070 PRINTTAB(1)"[SHIFTL]"TAB(26)"[SHIFT@]
"TAB(28)"[SHIFTL][2SPC][SHIFT@][3SPC]
[SHIFTL][2SPC][SHIFT@][CRSRD]"
1072 PRINTTAB(10)"[RVSON][A][RVSOFF]
"TAB(27)"*[SPC][RVSON][X][RVSOFF][SPC]
=[2SPC][RVSON][Y]"
1074 PRINT"[CRSRD]DONDE[SPC][RVSON][
A][RVSOFF][SPC]ES[SPC]LA[SPC]MATRIZ[SPC]
DE[SPC]COEFICIENTES,[SPC][RVSON][X][RVSOFF]
[SPC]ES[SPC]EL[SPC]VECTOR[SPC]DE";
1076 PRINT"[SPC]INCOGNITAS[SPC]E[SPC]
[RVSON][Y][RVSOFF][SPC]ES[SPC]EL[SPC]
VECTOR[SPC]DE[SPC]COEFICIENTES[SPC]I
NDEPENDIENTES";
1078 PRINT"[CRSRD][RVSON]PARA[SPC]CO
NTINUAR[SPC]PULSAR[RVSOFF][SPC]I"
1080 GETW$:IFW$<>"I"THEN1080
1082 PRINT"[CLR][4SPC]AL[SPC]EJECUTA
RSE[SPC]LA[SPC]SUBRUTINA[SPC]DE[SPC]
SISTEMAS[SPC]DE[SPC]ECUACIONES,[SPC]
SE[SPC]HA[SPC]";
1084 PRINT"DE[SPC]INTRODUCIR[SPC]COM
O[SPC]DATO[SPC]INICIAL[SPC]LA[SPC]DI
MENSION[SPC]DEL[SPC]SISTEMA,[SPC]O[SPC]
SEA,[SPC]";
1086 PRINT"EL[SPC]NUMERO[SPC]DE[SPC]
INCOGNITAS[SPC]QUE[SPC]CONTIENE[SPC]
EL[SPC]SISTEMA,[SPC]A[SPC]CONTINUACI
ON[SPC]";
1088 PRINT"SE[SPC]INTRODUCEN[SPC]LOS
[SPC]ELEMENTOS[SPC]DE[SPC]LA[SPC]PRI
MERA[SPC]FILA[SPC]DE[SPC]LA[SPC]MATR
IZ[SPC][RVSON][A][RVSOFF]";
1090 PRINT",[SPC]LUEGO[SPC]EL[SPC]PR
IMER[SPC]COEFICIENTE[SPC]INDEPENDIEN
TE[SPC]DEL[SPC]VECTOR[SPC][RVSON][Y]
[RVSOFF],[SPC]Y[SPC]";
1092 PRINT"ASI[SPC]SUCESIVAMENTE[SPC]
HASTA[SPC]COMPLETAR[SPC]TODAS[SPC]LA
S[SPC]FILAS[SPC]EXISTENTES."
1094 PRINT"[6CRSRD][RVSON]PARA[SPC]C
ONTINUAR[SPC]PULSAR[RVSOFF][SPC]I"
1096 GETW$:IFW$<>"I"THEN1096
1098 PRINT"[CLR][3SPC]TAMBIEN[SPC]PO
DRAS[SPC]RESOLVER[SPC]ECUACIONES[SPC]
POLINOMICAS[SPC]DE[SPC]COEFICIENTES"
;
1100 PRINT"[SPC]COMPLEJOS.[SPC]COMO[SPC]
POR[SPC]EJEMPLO[SPC]ESTA[SPC]SIGUIEN
TE:[CRSRD]"
1101 PRINTTAB(11)"4"TAB(23)"3"TAB(36
)"2"
1102 PRINT"[2SPC](3-5[RVSON]I[RVSOFF]
)*[SPC]X[SPC]+(-1+2[RVSON]I[RVSOFF])
*[SPC]X[SPC]+(.75+4[RVSON]I[RVSOFF])
*[SPC]X[SPC]+[CRSRD]"
1104 PRINTTAB(17)"1"TAB(29)"0"
1106 PRINT"[7SPC]+(8-2[RVSON]I[RVSOFF]
)*[SPC]X[SPC]+(.8-7[RVSON]I[RVSOFF])
*[SPC]X[SPC]=[SPC]0"
1108 PRINT"[CRSRD]DONDE[SPC][RVSON]I
[RVSOFF][SPC]REPRESENTA[SPC]EL[SPC]N
UMERO[SPC]COMPLEJO[SPC]DE[SPC]MODULO
[SPC]1[SPC]Y[SPC]ARGUMENTO";
1110 PRINT"[SPC]90[SPC]GRADOS.":PRIN
T"[4SPC]AL[SPC]EJECUTARSE[SPC]EL[SPC]
PROGRAMA,[SPC]EN[SPC]PRIMER[SPC]LUGA
R";
1112 PRINT"[SPC]SE[SPC]PREGUNTA[SPC]
EL[SPC]GRADO[SPC]DEL[SPC]POLINOMIO,[SPC]
A[SPC]CONTINUACION[SPC]HAY[SPC]QUE[SPC]
IR[SPC]";
1114 PRINT"ENTRANDO[SPC]LOS[SPC]DATO
S[SPC]CORRESPONDIENTES[SPC]A[SPC]LOS
[SPC]COEFICIENTES.[SPC]EN[SPC]TODO[SPC]
";
1116 PRINT"MOMENTO[SPC]SE[SPC]ESPECI
FICA[SPC]EN[SPC]LA[SPC]PANTALLA,[SPC]
CLARAMENTE,[SPC]QUE[SPC]COEFICIENTE"
;
1118 PRINT"[SPC]HAY[SPC]QUE[SPC]INTR
ODUCIR[SPC]PARA[SPC]QUE[SPC]NO[SPC]E
XISTA[SPC]POSIBILIDAD[SPC]DE[SPC]ERR
OR.[SPC]";
1119 PRINT"TAMBIEN,[SPC]DEBERAS[SPC]
INTRODUCIR[SPC]EL[SPC]ERROR[SPC]ADMI
SIBLE[SPC]EN[SPC]LAS[SPC]SOLUCIONES.
"
1120 PRINT"[CRSRD][RVSON]PARA[SPC]CO
NTINUAR[SPC]PULSAR[RVSOFF][SPC]I"
1122 GETW$:IFW$<>"I"THEN1122
1124 PRINT"[CLR][6CRSRD][4SPC]ADEMAS
,[SPC]ESTE[SPC]PROGRAMA[SPC]TE[SPC]P
ERMITE[SPC]REDONDEAR[SPC]LOS[SPC]";
1126 PRINT"RESULTADOS[SPC]HASTA[SPC]
LOS[SPC]DECIMALES[SPC]QUE[SPC]DESEES
.[SPC]ES[SPC]ACONSEJABLE[SPC]";
1128 PRINT"REDONDEAR[SPC]LOS[SPC]RES
ULTADOS,[SPC]YA[SPC]QUE[SPC]DE[SPC]E
STA[SPC]MANERA[SPC]SON[SPC]MAS[SPC]C
LAROS,[SPC]";
1130 PRINT"AUNQUE[SPC]ESTO[SPC]CONLL
EVA[SPC]UN[SPC]TIEMPO[SPC]DE[SPC]EJE
CUCION[SPC]MAS[SPC]LARGO."
1132 PRINT"[4CRSRD][RVSON]PARA[SPC]E
MPEZAR[SPC]CON[SPC]LA[SPC]EJECUCION[SPC]
DEL[SPC]PROGRAMA[SPC]PULSA[RVSOFF][SPC]
I"
1134 GETW$:IFW$<>"I"THEN1134
1136 RETURN
1300 END
```



Minas

Juan Clapes Ribas
Madrid, 35-5º-2
IBIZA (Baleares)

VIC-20
sin ampliación



El juego "Minas" consiste en atravesar un campo de minas sin chocar con ninguna de ellas ni con los bordes. Para eso tu VIC te ayuda diciéndote cuántas minas tienes a tu alrededor. Si te chocas, el ordenador te ofrece la repetición con las minas al descubierto.

El programa es muy sencillo y aquí está su explicación:

Línea 80 Coloca las minas.
Línea 90-120 Marco.
Línea 130-260 Recoge pulsación, comprueba si hay choque o si se ha llegado a la meta, dice el número de minas que hay alrededor, hace el movimiento.
Líneas 300-330 Repetición.
Líneas 500-515 Principio programa.
Líneas 600-680 Sonido.

```
1 REM JUAN CLAPES.
2 REM IBIZA
10 GOSUB500
15 GETA$:IFA$=""THEN15
70 DIME(200):PRINT"[10CRSRD]ELIGE[SPC]
NIVEL[SPC]DE[SPC]DIFI-[5SPC]CULTAD[SPC]
(1-5)":INPUTB:W=128
75 PRINT"[CLR][BLK]":POKE36879,30:PO
KE36878,15:S4=36877:S1=36874
80 FORQ=1TOB*75/2:M=INT(7680+506*RND
(1)):POKEM,42:NEXT
90 FORO=7724TO8164STEP22:POKEO,35:PO
KE30720+O,0:NEXT:FORO=8165TO8173
100 POKEO,35:POKE30720+O,0:NEXT:FORO
=8177TO8185:POKEO,35:POKE30720+O,0:N
EXT
110 FORO=8163TO7745STEP-22:POKEO,35:
POKE30720+O,0:NEXT:FORO=7744TO7737ST
EP-1
115 POKEO,35:POKE30720+O,0:NEXT:FORO
=7733TO7725STEP-1
120 POKEO,35:POKE30720+O,0:NEXT:X=81
75:O=0:POKE7734,25:POKE7735,25:POKE7
736,25
130 POKE8174,32:POKE8175,32:POKE8176
,32:PRINT"[HOM][RVSON]MINAS[SPC]VECI
NAS"
135 GETB$:IFB$=""THEN135
140 IFB$="I"THENT=-22:GOTO180
150 IFB$="J"THENT=-1:GOTO180
160 IFB$="L"THENT=1:GOTO180
170 IFB$="[SPC]"THENT=22:GOTO180
175 GOTO135
180 C=X+T:D=PEEK(C)
185 Z=Z+1:E(Z)=C
187 IFD=42THEN300
190 IFPEEK(C+1)=42THENR=R+1
200 IFPEEK(C-1)=42THENR=R+1
210 IFPEEK(C-22)=42THENR=R+1
220 PRINT"[HOM]"TAB(15)"[RVSON]"R:R=
0
240 IFD=35THEN300
250 IFD=25THEN400
255 POKEC,65:POKEX,96+128:POKE30720+
X,2:POKEC+30720,6
260 X=C:GOSUB670:GOTO135
300 GOSUB650:POKEC,83:POKE36879,56:F
ORG=Z-1TO0STEP-1
310 F=E(G):POKEF,96+128:POKE30720+F,
7
320 FORO=1TO200:NEXT:NEXT
330 FORG=1TOZ:POKEE(G),65:POKEE(G-1)
,224:POKE30720+E(G),1:POKE30720+E(G-
1),1
340 GOSUB600:NEXT:PRINT"[6CRSRD][CRSRR]
[BLK]LO[SPC]SIENTO,HAS[SPC]VOLADO":G
OTO630
345 PRINT"[HOM][9CRSRD][CRSRR]OTRA(S
/N)":GETA$:IFA$=""THEN345
350 IFA$="S"THENCLR:GOTO10
360 IFA$="N"THENEND
370 GOTO345
400 POKE36879,56:PRINT"[6CRSRD][CRSRR]
LO[SPC]HAS[SPC]HECHO[SPC]MUY[SPC]BIE
N":GOTO630
500 POKE36879,8:PRINT"[CLR][5CRSRD][2CRSRR]
[WHT]*[SPC]*[SPC]*[SPC]*[2SPC]*[SPC]
***[SPC]***":PRINT"[2CRSRR]***[SPC]*
[SPC]**[SPC]*[SPC]*[SPC]*[SPC]*"
510 PRINT"[2CRSRR]*[SPC]*[SPC]*[SPC]
*[SPC]**[SPC]***[SPC]***":PRINT"[2CRSRR]
*[SPC]*[SPC]*[SPC]*[2SPC]*[SPC]*[SPC]
*[3SPC]*":PRINT"[2CRSRR]*[SPC]*[SPC]
*[SPC]*[2SPC]*[SPC]*[SPC]*[SPC]***
515 PRINT"[HOM][3CRSRD][6CRSRR]*
520 RETURN
600 POKES1,255:FORO=0TO399:NEXT:POKE
S1,0:RETURN
630 FORO=200TO255:POKES4,O:FORW=1TO2
0:NEXT:NEXT:POKES4,0:GOSUB345
650 FORO=255TO200STEP-1:POKES4,O:FOR
W=1TO20:NEXT:NEXT:POKES4,0:RETURN
670 W=W+1:IFW>255THENW=255
680 POKE36876,W:FORO=0TO20:NEXT:POKE
36876,0:RETURN
```



Directorio impreso

Tener una copia impresa del directorio de un disco es siempre útil. Para hacerlo teclea:

Load"$",8:open1,4:cmd1:list

Cuando termine, teclea:

Print#1:close1

T.S. RUN EE.UU.

Carreras-64

Félix García Gutiérrez
12 de Octubre, 16-Int. 6º B
VALLADOLID

C-64



Os mando como colaboración la adaptación para el CBM-64 hecha por mí del programa publicado en el nº 8 de la revista COMM.WORLD en el VIDEO-CASINO, para el VIC-20.
He introducido varias novedades que paso a comentaros:
En primer lugar he aumentado los caballos de 5 a 9; y también he puesto caracteres programados, dando un aspecto de caballo, y qué más quisiera yo que de caballo de carreras.
Otra novedad es que existe la posibilidad de que puedan jugar hasta CUATRO personas haciendo apuestas, todo esto se representa en la parte inferior de la pantalla.
Hay alguna cosilla más, pero creo sin importancia. El resto no hace falta añadir más ya que viene estupendamente "abrigado" en el nº 8 de la revista.
Hasta otra.

```
5 REM*CARRERAS**COMM.WORLD-8*****
6 REM**ADAPTADO PARA CBM-64 POR**
7 REM**FELIX GARCIA GUTIERREZ****
8 REM
10 PRINT"[CLR][12CRSRD][8CRSRR]UN[SPC]
MOMENTO[SPC]POR[SPC]FAVOR..."
11 PRINTCHR$(142):POKE52,48:POKE56,4
8:CLR
12 POKE56334,PEEK(56334)AND254:POKE1
,PEEK(1)AND251
13 FORI=0TO1023:POKEI+12288,PEEK(I+5
3248):NEXT:POKE1,PEEK(1)OR4
14 POKE56334,PEEK(56334)OR1:POKE5327
2,(PEEK(53272)AND240)+12
15 FORI=13048TO13055:READQ:POKEI,Q:N
EXT
16 FORI=13040TO13047:READW:POKEI,W:N
EXT
17 DATA1,3,3,191,223,22,36,18
18 DATA228,142,95,249,240,80,72,132
20 PRINT"[CLR][COMM3][10SPC]CARRERAS
[SPC]DE[SPC]CABALLOS[10SPC]"
21 PRINT"[3CRSRD][BLK][11CRSRR]MAXIM
O[SPC]4[SPC]JUGADORES[CRSRD]"
22 PRINT"N[COMMV]JUGADORES?":INPUTN:
FORA=1TON:R1(A)=500
23 PRINT"[CRSRD]NOMBRE[SPC]DEL[SPC]J
UGADOR[SPC]N[COMMV]"A:INPUTN$(A):NEX
TA
24 PRINT"[CLR][WHT]"
25 ML=1024:MN=55296:LA=40:POKE53280,
5:POKE53281,0
26 S1=54272
30 B=0:R=0:FORI=1TO9:A(I)=0:B(I)=0:N
EXTI
80 PRINT"BIENVENIDO[SPC]A[SPC]LAS[SPC]
CARRERAS[2SPC]"
85 FORI=1TO500:NEXT
86 X1=2:X2=2:X3=2:X4=2:X5=2:X6=2:X7=
2:X8=2:X9=2
100 PRINT"LAS[SPC]APUESTAS[SPC]ESTAN
[5SPC]4:1":PRINT
105 GOTO6000
400 PRINT"[CLR]";:GOSUB5500
420 FORL=1063TO1783STEP40:POKEL,160:
POKEL+54272,14:NEXTL
425 PRINT"[HOM][19CRSRD][COMM1][RVSON]
[40SPC][RVSOFF]";
427 GOSUB8000
430 GOSUB4000
600 R=INT(9*RND(1)+1)
620 GOSUB1000:GOSUB5000
625 ONRGOTO2010,2030,2050,2070,2090,
2110,2130,2150,2170
630 B(R)=B(R)+1
640 IFB(R)<37THEN600
655 GOSUB3000:FORT=1TO1000:NEXTT
665 PRINT"[CLR]":FORI2=1TOLA:PRINT"-
";:NEXTI2
667 GOTO7000
690 Z$="":PRINT"[CRSRD]OTRA[SPC]CARR
ERA?[SPC](S/N)";
691 INPUTZ$
700 IFZ$="S"THEN24
710 POKE53280,14:POKE53281,6:PRINT"[CLR]
[COMM7]";:END
990 END
1000 REM*SUMA AL CONTADOR APROPIADO*

1090 IFR=1THENX1=X1+1
1092 IFR=2THENX2=X2+1
1093 IFR=3THENX3=X3+1
1094 IFR=4THENX4=X4+1
1095 IFR=5THENX5=X5+1
1096 IFR=6THENX6=X6+1
1097 IFR=7THENX7=X7+1
1098 IFR=8THENX8=X8+1
1099 IFR=9THENX9=X9+1
1100 RETURN
2000 REM*COLOCA CABALLOS EN PANTALLA

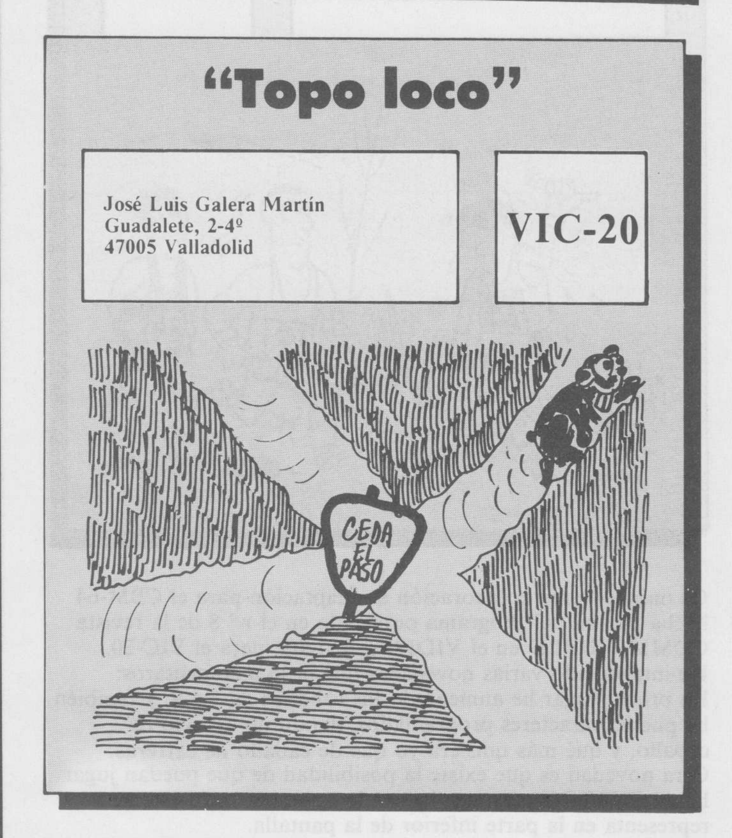
2010 Y=1
2020 POKEML+X1+40*Y,94:POKEMN+X1+40*
Y,1
2021 POKEML+X1-1+40*Y,95
2022 POKEML+X1-2+40*Y,32
2023 GOTO630
2030 Y=3
2040 POKEML+X2+40*Y,94:POKEMN+X2+40*
Y,2
2041 POKEML+X2-1+40*Y,95
2042 POKEML+X2-2+40*Y,32
2043 GOTO630
2050 Y=5
2060 POKEML+X3+40*Y,94:POKEMN+X3+40*
Y,3
2061 POKEML+X3-1+40*Y,95
2062 POKEML+X3-2+40*Y,32
2063 GOTO630
2070 Y=7
2080 POKEML+X4+40*Y,94:POKEMN+X4+40*
Y,4
2081 POKEML+X4-1+40*Y,95
2082 POKEML+X4-2+40*Y,32
2083 GOTO630
2090 Y=9
```

```
2100 POKEML+X5+40*Y,94:POKEMN+X5+40*
Y,5
2101 POKEML+X5-1+40*Y,95
2102 POKEML+X5-2+40*Y,32
2103 GOTO630
2110 Y=11
2120 POKEML+X6+40*Y,94:POKEMN+X6+40*
Y,6
2121 POKEML+X6-1+40*Y,95
2122 POKEML+X6-2+40*Y,32
2123 GOTO630
2130 Y=13
2140 POKEML+X7+40*Y,94:POKEMN+X7+40*
Y,7
2141 POKEML+X7-1+40*Y,95
2142 POKEML+X7-2+40*Y,32
2143 GOTO630
2150 Y=15
2160 POKEML+X8+40*Y,94:POKEMN+X8+40*
Y,8
2161 POKEML+X8-1+40*Y,95
2162 POKEML+X8-2+40*Y,32
2163 GOTO630
2170 Y=17
2180 POKEML+X9+40*Y,94:POKEMN+X9+40*
Y,9
2181 POKEML+X9-1+40*Y,95
2182 POKEML+X9-2+40*Y,32
2183 GOTO630
3000 REM*SONIDO*
3010 POKE54296,9:POKES1+5,125:POKES1
+1,43:POKES1,52
3020 POKES1+4,33:FORT=1TO25:NEXTT
3030 POKE54296,0:POKES1+5,0:RETURN
4000 POKE54296,15:S1=54272:REM*SONID
O PARA COMENZAR*
4010 FORY=0TO18:POKEML+3+40*Y,115:PO
KEML+54272+3+40*Y,14:NEXTY
4020 FORJ2=1TO1000:NEXTJ2
4040 FORX=15TO0STEP-1:POKE54296,X:PO
KES1+4,129:POKES1+5,15
4050 POKES1+1,40:POKES1,200:NEXTX
4100 POKE54296,0:POKES1+5,0:RETURN
5000 POKE54296,14:POKES1+1,48:POKES1
,127:FORT=1TO25:NEXT
5001 POKE54296,0:RETURN
5500 A=1:B=1
5510 FORY=1TO17STEP2
5520 POKEML+2+40*Y,94:POKEML+54272+2
+40*Y,B
5525 POKEML+1+40*Y,95:POKEML+1+54272
+40*Y,B
5530 POKEML+40*Y,48+A:POKEML+54272+4
0*Y,B
5540 A=A+1:B=B+1
5550 NEXT:RETURN
6000 FORA=1TON
6022 IFR1(A)=0THENPRINTN$(A);"[SPC]N
O[SPC]TIENE[SPC]DINERO":R(A)=0:H(A)=
0:GOTO6045
6025 PRINT"EL[SPC]JUGADOR[SPC]N[COMMV]
"A;N$(A);"[SPC]APUESTA[SPC]POR[SPC]E
L[SPC]CABALLO[SPC]N[COMMV]":INPUTH(A
)
6030 IFH(A)<1ORH(A)>9THEN6025
6035 PRINT"CUANTO[SPC]APUESTAS?":INP
UTR(A)
6040 IFR(A)<1ORR(A)>R1(A)THENPRINT"[3CRSRU]
":GOTO6035
6045 NEXTA
6050 GOTO400
7000 PRINT"[HOM][2CRSRD][CRSRR]":FOR
A=1TON
7005 IFH(A)=RTHENR1(A)=R1(A)+(4*R(A)
):GOTO7015
7010 R1(A)=R1(A)-R(A)
7015 PRINT"APOSTANTE[SPC]N[COMMV]"A;
N$(A);TAB(30)R1(A)
7020 NEXTA
7025 GOTO690
8000 W=1:PRINT"[HOM][20CRSRD]"
8010 FORA=1TON
8020 PRINTTAB(W)N$(A):PRINTTAB(W)R1(
A):PRINTTAB(W-1)R(A)"/"H(A)
8030 PRINT"[5CRSRU]"
8040 W=W+10:NEXTA:RETURN
```

"Topo loco"

José Luis Galera Martín
Guadalete, 2-4º
47005 Valladolid

VIC-20



Esta nueva versión del "Topo Loco" es el resultado de un estudio detallado de la elaborada por Manuel Torralba que publicó Commodore World en su número de marzo de este año.

El listado es completamente nuevo y proporciona al juego el atractivo del color ya que las lechugas son de color verde, los bloques, letras y números de color azul, el rastro del topo y su explosión de color púrpura y el huerto de un color terroso claro enmarcado en azul verdoso claro que se obtiene con el número 27 del código de colores de pantalla y marco. La diferencia de color entre el huerto, simulado por la pantalla, y el marco permite distinguir con nitidez los límites del huerto.

Con objeto de evitar la repetición del texto inicial para jugar de nuevo he introducido las líneas 560 y 570 que mediante la pulsación del espaciador ponen en marcha el programa en la línea 120 y ahorran un tiempo precioso al jugador impaciente.

Los límites del huerto pueden fijarse a voluntad según cuatro modalidades que se ofrecen en un menú previo al inicio del juego y que permiten limitar el huerto por los cuatro costados, por los costados horizontales, por los costados verticales o por ningún costado. El lector se hará una idea perfecta de cómo funciona cada modalidad sabiendo que si el huerto está limitado por los costados horizontales el topo pasa del costado derecho al izquierdo o viceversa manteniendo la fila y la dirección de movimiento pero explota si intenta salir del huerto por el costado superior o por el inferior.

Para que la dirección del topo sea más fácil e intuitiva he sustituido la I, la J, la L y el espaciador por la P, la D, la F y la L respectivamente de modo que con la mano izquierda se controlen los movimientos horizontales y con la derecha los verticales.

Las instrucciones de las líneas 250 y 280 evitan los problemas

que ocasiona la superposición de bloques y lechugas mediante un rastreo previo de la posición de pantalla en la que se va a colocar un bloque o una lechuga. Gracias a estas dos líneas el número de bloques del primer huerto es quince y aumenta en cinco de un huerto al siguiente y el número de lechugas es siempre quince. Lo primero es interesante para aumentar la dificultad y estimular al jugador, pero lo segundo es esencial para la buena marcha del juego. Si el número de lechugas fuese inferior a quince como consecuencia de la superposición de lechugas, el topo continuaría moviéndose tras engullirlas todas y no habría forma de pasar al huerto siguiente. El jugador sólo tendría entonces dos alternativas: parar el programa y ponerlo de nuevo en marcha o dejar que explotasen uno tras otro los topos que le quedasen hasta consumir los cuatro de que dispone y ambas suponen comenzar de nuevo en el huerto uno. El aumento de velocidad del topo es mucho más notorio en esta versión del juego gracias al nuevo procedimiento empleado que en lugar de restar una cantidad fija en cada movimiento a la variable del bucle de retardo que controla la velocidad y partir en todos los huertos del mismo valor inicial la mantiene constante en cada huerto y le resta el diez por ciento de su valor al pasar de cada huerto al siguiente.

La última novedad del programa que merece comentario es que he sustituido siempre que he podido las instrucciones condicionales por fórmulas que utilizan la capacidad del VIC-20 de evaluar con -1 las proposiciones lógicas verdaderas y con 0 las falsas. Esta sustitución abrevia notablemente el programa y proporciona una mayor rapidez en su ejecución. Una muestra de esta técnica puede verse en la línea 330 que sustituye a las líneas 25, 30, 40, 60 y 70 y en la línea 340 que realiza la misma función que las líneas 80, 90, 100 y 110.

```
 1 REM********************
 2 REM*     TOPO  LOCO     *
 3 REM*------------------*
 4 REM*        POR         *
 5 REM*  J.  L.  GALERA  *
 6 REM********************
10 PRINT"[CLR]":POKE36879,25:POKE368
78,15:POKE36865,150
20 PRINT"[3CRSRD][4CRSRR]**TOPO[SPC]
LOCO**"
30 PRINT"[2CRSRD][2CRSRR]MUEVE[SPC]A
L[SPC]TOPO[SPC]LOCO"
40 PRINT"[CRSRD][3CRSRR]Y[SPC]COME[SPC]
TODAS[SPC]LAS"
50 PRINT"[CRSRD]LECHUGAS[SPC]([GRN][SHIFTX]
[BLU]),[SPC]EVITANDO"
60 PRINT"[CRSRR]CHOCAR[SPC]CON[SPC]T
U[SPC]RASTRO"
70 PRINT"[CRSRD][CRSRR]O[SPC]LOS[SPC]
BLOQUES[SPC]AZULES."
80 PRINT"[2CRSRR]DIRIGE[SPC]AL[SPC]T
OPO[SPC]CON"
90 PRINT"[CRSRD][7CRSRR]D,F,L,P."
100 FORAL=150TO36STEP-0.07:POKE36865
,AL:NEXTAL
110 FORI=1TO3000:NEXTI
120 PRINT"[CLR][4CRSRD][2SPC]ELIGE[SPC]
ENTRE[SPC]LAS[SPC]SI-[2SPC]GUIENTES[SPC]
MODALIDADES:"
130 PRINT"[CRSRD][SPC]1:HUERTO[SPC]C
ERCADO[SPC]POR[4SPC]TODAS[SPC]PARTES
."
140 PRINT"[SPC]2:HUERTO[SPC]CERCADO[SPC]
POR[4SPC]LA[SPC]IZQUIERDA[SPC]Y[SPC]
POR[4SPC]LA[SPC]DERECHA."
150 PRINT"[SPC]3:HUERTO[SPC]CERCADO[SPC]
POR[4SPC]ARRIBA[SPC]Y[SPC]POR[SPC]AB
AJO."
160 PRINT"[CRSRU][SPC]4:HUERTO[SPC]S
IN[SPC]CERCAR.":INPUT"[2CRSRD][5SPC]
MODALIDAD=";E
170 IFE<>1ANDE<>2ANDE<>3ANDE<>4THEN1
20
180 PRINT"[CLR][6CRSRD]ELIGE[SPC]DIF
ICULTAD[SPC]ENTRE[CRSRD]1[SPC]Y[SPC]
20.[SPC][SHIFTL]A[SPC]MAYOR[SPC]DIFI
-"
190 PRINT"CULTAD[SPC]CORRESPONDE[SPC]
AL[SPC][CRSRD][10CRSRR]1.[4CRSRD][4CRSRL]
D=";
200 INPUTD
210 IFD<1ORD>20THEN180
220 H=1:T=4:B=15:V=30*D:POKE36879,27

230 L=15:PRINT"[CLR][11CRSRD][6CRSRR]
HUERTO";H:FORI=1TO1000:NEXTI
240 FORI=1TOB
250 K=INT(RND(1)*506):IFPEEK(7680+K)
=102THEN250
260 POKE7680+K,102:POKE38400+K,6:NEX
TI
270 FORI=1TO15
280 K=INT(RND(1)*506):IFPEEK(7680+K)
=102ORPEEK(7680+K)=88THEN280
290 POKE7680+K,88:POKE38400+K,5:READ
A:POKE36876,A:FORJ=1TO150:NEXT:NEXT
300 POKE36876,0:RESTORE
310 FORI=1TO1000:NEXTI
320 X=12:Y=11:M=0
330 GETT$:M=M*(T$<>"D")*(T$<>"F")*(T
$<>"P")*(T$<>"L")-(T$="L")-2*(T$="F"
)-3*(T$="P")
340 XP=X+(M=0)-(M=2):YP=Y-(M=1)+(M=3
)

350 P1=22*Y+X:P2=22*YP+XP:CA=PEEK(76
80+P2)
360 IFXP<0ORXP>21ORYP<0ORYP>22THENON
EGOTO430,580,600,620
370 IFCA=102ORCA=81ORCA=42THEN430
380 X=XP:Y=YP:POKE7680+P2,81:POKE384
00+P2,4
390 IFCA<>88THENPOKE36876,150:GOTO42
0
400 FORI=150TO250:POKE36876,I:NEXTI:
POKE36876,0:L=L-1
410 IFL=0THEN480
420 FORI=1TOV:NEXT:POKE36876,0:GOTO3
30
430 POKE7680+P1,42:POKE38400+P1,4:C=
256
440 FORI=15TO0STEP-0.1:C=C-1:POKE368
79,C:POKE36878,I:POKE36877,180:NEXTI

450 POKE36877,0:POKE36878,15:POKE368
79,27:T=T-1
460 IFT=0THEN510
470 GOTO310
480 FORI=1TO15:READA:NEXTI
490 FORI=1TO31:READA,A1:POKE36876,A:
POKE36879,A:FORJ=1TOA1:NEXTJ:POKE368
76,0:NEXTI
500 RESTORE:H=H+1:B=B+5:V=V-V/10:GOT
O230
510 FORI=1TO77:READA:NEXTI
520 FORI=1TO16:READA:POKE36876,A:FOR
J=1TO150:NEXTJ:NEXTI:RESTORE
530 POKE36879,25:PRINT"[CLR][9CRSRD]
[4CRSRR]FIN[SPC]DEL[SPC]JUEGO."
540 PRINT"[CRSRD][CRSRR]PARA[SPC]JUG
AR[SPC]DE[SPC]NUEVO"
550 PRINT"[CRSRD][CRSRR]PULSA[SPC]EL
[SPC]ESPACIADOR."
560 GETA$:IFA$<>"[SPC]"THEN560
570 RUN120
580 IFXP<0ORXP>21THEN430
590 YP=-22*(YP<0):GOTO350
600 IFYP<0ORYP>22THEN430
610 XP=-21*(XP<0):GOTO350
620 XQ=-21*(XP<0)-XP*(YP<0)-XP*(YP>2
2):YP=-YP*(XP<0)-YP*(XP>21)-22*(YP<0
):XP=XQ:GOTO350
630 DATA195,201,207,209,215,219,223,
225,223,219,215,209,207,201,195
```

```
640 DATA195,150,209,150,209,150,215,
150,219,150,209,150,219,150,215,150
650 DATA195,150,209,150,209,150,215,
150,219,150,209,300,207,150,195,150,
209,150,209,150
660 DATA215,150,219,150,221,150,219,
150,215,150,209,150,207,150,195,150
670 DATA201,150,207,150,209,300,209,
150,27,150
680 DATA225,223,0,219,215,0,209,207,
201,207,201,195,0,0,225,0
```

Ahorcado

Rogelio Douton
Entenza, 6 - 5º 2º
08015 BARCELONA

C-64



Este es el típico juego del ahorcado. Un jugador escribe una palabra o una frase (no mayor de 2 líneas de pantalla) y el otro sin haberla visto, claro está, debe adivinarla pulsando las letras que crea convenientes, tiene 6 intentos para lograrlo. A cada letra pulsada que no este en la palabra se le va dibujando un trozo de horca, donde quedará colgado él si comete más de 6 fallos. Si logra adivinarla sale una pantalla de felicitación. Por propia experiencia sé que es un juego muy apto para niños y para reuniones de adultos de 4 ó 5 personas.

A la hora de teclear el programa lo más engorroso serán las sentencias que se encargan de los dibujos, en concreto, de la 470 a la 630 (dibujo de cuando estas ahorcado) y de la 1000 a la 2400 dibujo del proceso de ahorcamiento) y de la 8000 a la 12500 (pantalla de felicitación), podéis poner el dibujo que queráis inventaros, os aconsejo primero hacer el dibujo (con los signos de las teclas frontales) y después poner el número de sentencia, el PRINT, el TAB y abrir comillas (no es preciso cerrarlas), luego pulsar RETURN, y lo mismo con la siguiente línea.

Está diseñado para televisión en color, pero en blanco y negro también se ve bien. Tiene una rutina de sonido para cada vez que pulsas una tecla.

Al jugar es importante acordarse de las letras pulsadas erróneas para no volver a repetirlas.

La elección de las combinaciones de colores es obra de mi compañera; así pues, el programa también tiene un delicado toque femenino. Que os divirtáis!

```
10 PRINT"[CLR]":Z=0:DN=0:POKE53280,7
 :POKE53281,10:PRINTCHR$(144)
12 PRINT"QUIERES[SPC]INSTRUCIONES?"
13 PRINT"SI[SPC]LAS[SPC]QUIERES[SPC]
PULSA[2SPC]S"
14 PRINT"SI[SPC]NO[SPC]LAS[SPC]QUIER
ES[SPC]PULSA[2SPC]N"
15 GETH$:IFH$=""THEN15
16 CS=160:GOSUB15300
17 IFH$="N"THEN120
20 GOSUB5000
29 POKE53280,13:POKE53281,8:PRINTCHR
$(144)
30 PRINT"YA[SPC]HABEIS[SPC]DECIDIDO[SPC]
QUIEN[SPC]EMPIEZA?"
40 FORX=1TO2000:NEXT:PRINT"[CLR]"
50 PRINT"BIEN,ESCRIBE[SPC]LA[SPC]PAL
ABRA[SPC]O[SPC]LA[SPC]FRASE[SPC]QUE[2SPC]
QUIERAS,Y[SPC]LUEGO[SPC]PULSA[SPC]LA
[SPC]TE";
60 PRINT"CLA[SPC]RETURN,TUOPONENTE[SPC]
TIENE[SPC]6[2SPC]OPORTUNIDADES[SPC]P
ARA[SPC]ADIVINAR[SPC]LO[SPC]QUE";
70 PRINT"[SPC]HAS[SPC]ESCRITO,INCLUI
DOS[SPC]LOS[2SPC]ESPACIOS[SPC]ENTRE[SPC]
PALABRAS,SI[SPC]HAY[SPC]UNA[SPC]FRAS
E"
75 PRINT:PRINT"LOS[SPC]ESPACIOS[SPC]
SE[SPC]MARCAN[SPC]PULSANDO[SPC]LA[SPC]
BARRAESPACIADORA."
78 PRINT"LA[SPC]FRASE[SPC]NO[SPC]PUE
DE[SPC]ESCEDER[SPC]DE[SPC]DOS[SPC]LI
NEAS."
80 PRINT:PRINT"DE[SPC]ACUERDO?,[SPC]
PARA[SPC]COMENZAR[SPC]APRIETA[SPC]LA
[SPC]TECLA[2SPC]S."
90 GETF$:IFF$=""THEN90
95 CS=160:GOSUB15300
100 IFF$="S"GOTO120
110 GOTO90
120 GOSUB5000
130 POKE53280,13:POKE53281,4 :PRINTC
HR$(144):INPUTA$
140 GOSUB5000
150 PRINT"COMIENZA[SPC]A[SPC]TECLEAR
[SPC]LETRAS[SPC]DE[SPC]UNA[SPC]EN[SPC]
UNA"
160 FORX=1TO2000:NEXT:PRINT"[CLR]"
162 PRINT"[HOM]":FORY=1TOLEN(A$)
164 PRINTTAB(1)".";
166 NEXT
170 GETC$:IFC$=""GOTO170
180 CN=0:X=0:Y=0
182 CS=170:GOSUB15300
185 FORY=1TOLEN(G$):IFMID$(G$,Y,1)=C
$THENGOTO170
186 NEXTY
190 FORX=1TOLEN(A$)
200 B$=MID$(A$,X,1)
202 IFB$=C$THENF$=B$:G$=G$+F$
210 IFB$=C$THENPRINT"[HOM]":PRINTTAB
(X)B$:PRINT"[2CRSRU]":CN=CN+1:DN=DN+
1
215 IFB$=C$THENCS=250:GOSUB15300
220 IFDN=LEN(A$)THEN7490
230 NEXT
240 IFCN<>0GOTO170
250 Z=Z+1:CS=140:GOSUB15300
260 ONZGOTO950,1250,1550,1850,2150
270 IFZ=6GOTO450
280 GOTO170
450 PRINT"[CLR]":POKE53280,0:POKE532
81,2
460 FORY=1TO500:NEXT
470 PRINT:PRINT:PRINT:PRINT
500 PRINT"[20SPC][SHIFTO][COMMY][SHIFT£]
[3COMMY][SHIFTH]
510 PRINT"[20SPC][COMMH][SHIFT£][4SPC]
[SHIFTH]
520 PRINT"[20SPC][SHIFT£][5SPC]O
```

```
530 PRINT"[20SPC][COMMH][5SPC][SHIFTB]

540 PRINT"[20SPC][COMMH][4SPC][SHIFTN]
[SHIFTB][SHIFTM]
550 PRINT"[20SPC][COMMH][5SPC][SHIFTB]

560 PRINT"[20SPC][COMMH][4SPC][SHIFTN]
[SPC][SHIFTM]
570 PRINT"[20SPC][COMMH][3SPC][SHIFTN]
[3SPC][SHIFTM]
580 PRINT"[20SPC][COMMH][2SPC][SHIFTJ]
[5SPC][SHIFTK]
590 PRINT"[20SPC][COMMH]
600 PRINT"[20SPC][COMMH]
610 PRINT"[20SPC][SHIFTL][COMMP]
620 PRINT"
630 PRINT"[17SPC]LA[SPC]VIDA[SPC]ES[SPC]
CRUEL!
632 FORX=1TO1500:NEXT
635 PRINT"[CRSRD][PUR][2SPC]ESTO[SPC]
ERA[SPC]LO[SPC]MISTERIOSO:[SPC]":PRI
NT"[BLK]":PRINT"[2SPC]"A$
640 END
950 PRINT"[4CRSRD]"
1000 PRINTTAB(22)       "[7COMMP]
1100 PRINTTAB(22)       "[COMMK][2SPC]
[SHIFT£][3SPC][SHIFTT]
1200 PRINTTAB(22)       "[COMMK][SPC]
[SHIFT£][4SPC][SHIFTT]
1202 GOTO3000
1250 PRINT"[7CRSRD]"
1300 PRINTTAB(22)       "[COMMK][SHIFT£]
[5SPC]O
1400 PRINTTAB(22)       "[COMMK][5SPC]
[SHIFTN][COMMT][SHIFTM]
1500 PRINTTAB(22)       "[COMMK][4SPC]
[COMMM][SHIFTH][SPC][SHIFTG][COMMG]
1502 GOTO3000
1550 PRINT"[10CRSRD]"
1600 PRINTTAB(22)       "[COMMK][4SPC]
[SHIFTJ][SHIFTH][COMMG][SHIFTG][SHIFTK]

1700 PRINTTAB(22)       "[COMMK][5SPC]
[COMMM][SPC][COMMG]
1800 PRINTTAB(22)       "[COMMK][5SPC]
[COMMM][SPC][COMMG]
1802 GOTO3000
1850 PRINT"[13CRSRD]"
1900 PRINTTAB(22)       "[COMMK][5SPC]
[SHIFT@][SPC][SHIFTL]
2000 PRINTTAB(22)       "[COMMK][SPC]
[COMML][9COMMU][COMMK]
2100 PRINTTAB(22)       "[COMMK][SPC]
[COMML][9SPC][COMMK]
2102 GOTO3000
2150 PRINT"[16CRSRD]"
2200 PRINTTAB(22)       "[COMMK][SPC]
[COMML][9SPC][COMMK]
2300 PRINTTAB(22)       "[17COMMY]
2400 PRINTTAB(20)    "A[SPC]LA[SPC]PR
OXIMA...ZAS!"
2401 FORX=1TO11:GOSUB15300:NEXT
2402 GOTO3000
3000 PRINT"[HOM]":GOTO170
4300 PRINT:PRINT
5000 FORH=1TO1000:NEXT:PRINT"[CLR]":
PRINT"[8CRSRD]":RETURN
7490 FORP=1TO300:NEXT:POKE53280,4
7500 PRINT"[CLR]":POKE53281,14
8000 PRINTTAB(21)      "[7SPC].
8100 PRINTTAB(21)      "[7SPC].
8200 PRINTTAB(21)      "[7SPC].
8300 PRINTTAB(21)      "[2SPC][SHIFTM][3SPC]
[SHIFTN].
8400 PRINTTAB(21)      "[3SPC][SHIFTM][SPC]
[SHIFTN][SPC].[SPC][4SPC]
8500 PRINTTAB(21)      "[3SPC][SHIFTJ][SHIFTQ]
[2SPC].[5SHIFTN]
8600 PRINTTAB(21)      "[3SPC][SHIFTW][SPC]
[SHIFTW][SPC].[SHIFTW][SHIFTB][SHIFTW]
[SPC][COMMH][SHIFTI]
8700 PRINTTAB(21)      "[3SPC][3SHIFTJ]
[SPC].[COMMH][SHIFTJ][SHIFTU][SHIFTK]
[COMMH][SHIFTK][SPC]
8800 PRINTTAB(21)      "[4SPC][2COMMH][SPC]
.[COMMH][SHIFTC][SHIFTK][SPC][COMMH]

8900 PRINTTAB(21)      "[4SPC][2COMMH][SPC]
.[SHIFTM][2SPC][SHIFTN].
9000 PRINTTAB(21)      "[4SPC][2COMMH][2SPC]
.[3SPC].
9100 PRINTTAB(21)      "[4SPC][COMMH][SHIFTM]
[SPC][2SPC]...[SPC]
9200 PRINTTAB(21)      "[4SPC][SHIFTM][SPC]
[SHIFTM]
9300 PRINTTAB(21)      "[5SPC][SHIFTM][SPC]
[SHIFTM]
9400 PRINTTAB(21)      "[6SPC][SHIFTM][9SPC]

9500 PRINTTAB(21)      "[7SPC][SHIFTM][9SPC]

10000 POKE53281,6:FORJ=1TO300:NEXT:P
OKE53281,14:PRINT"[CLR]"
11000 PRINTTAB(22)            "[6SPC].
11100 PRINTTAB(22)            "
11200 PRINTTAB(22)            "[6SPC][SHIFTN]

11300 PRINTTAB(22)            "[5SPC][SHIFTN]
.
11400 PRINTTAB(22)            "[4SPC][SHIFTN]
[SPC].[SPC][5SHIFTN]
11500 PRINTTAB(22)            "[2SPC][SHIFTJ]
[SHIFTQ][2SPC].[5SHIFTN]
11600 PRINTTAB(22)            "[2SPC][2SHIFTW]
[2SPC].[SHIFTQ][SHIFTB][SHIFTQ][SPC]
[COMMH][SHIFTI]
11700 PRINTTAB(22)            "[2SPC][3SHIFTJ]
[SPC].[COMMH][SHIFTJ][2SPC][COMMH][SHIFTK]
[SPC]
11800 PRINTTAB(22)            "[3SPC][2COMMH]
[SPC].[COMMH][SHIFTC][SHIFTK][SPC][COMMH]

11900 PRINTTAB(22)            "[3SPC][2COMMH]
[SPC].[SHIFTM][2SPC][SHIFTN].
12000 PRINTTAB(22)            "[3SPC][2COMMH]
[2SPC].[3SPC].
12100 PRINTTAB(22)            "[3SPC][COMMH]
[SHIFTM][SPC][2SPC]...[SPC]
12200 PRINTTAB(22)            "[3SPC][SHIFTM]
[SPC][SHIFTM]
12300 PRINTTAB(22)            "[4SPC][SHIFTM]
[SPC][SHIFTM]
12400 PRINTTAB(22)            "[5SPC][SHIFTM]
[9SPC]
12500 PRINTTAB(22)            "[7SPC][9SPC]

12525 POKE53281,6
12550 IFAA=2THENGOTO12700
12600 FORJ=1TO300:NEXT:PRINT"[CLR]":
AA=AA+1:GOTO7500
12700 PRINT"[WHT]":IFZ=0GOTO14000
12800 IFZ=5GOTO15000
12900 PRINT"[HOM]";"[7CRSRD][3CRSRR]
!BRAVO[SPC]COLEGA!
13000 PRINTSPC(17)"[11CRSRD]!SOLO[SPC]
";:PRINTZ;:PRINT"[SPC]FALLOS!"
13100 END
14000 PRINT"[HOM]";"[7CRSRD][3CRSRR]
!ESTUPENDO[SPC]ARTISTA!
14100 PRINTSPC(17)"[11CRSRD]NI[SPC]
UN[SPC]SOLO[SPC]FALLO!"
14200 END
15000 PRINT"[HOM]";"[7CRSRD][5CRSRR]
UF[SPC]!!..."
15100 PRINTSPC(11)"[11CRSRD]!TE[SPC]
SALVASTE[SPC]POR[SPC]LOS[SPC]PELOS!"

15200 END
15300 W=54272:P=1:WF=W+4:V=W+24:AT=W
+5:SE=W+6:HF=W+1:LF=W:POKEV,15
```

```
15310 POKEAT,9:POKESE,0:POKEHF,CS-10
0:POKELF,CS:POKEWF,17
15320 FORT=1TO50:NEXT:POKEWF,16:FORJ
=WTOW+23:POKEJ,0:NEXT:RETURN
```

Caza gatos

Daniel Audi Alberich
Prol. Rambla Cataluña, s/n
Telf.: (977) 501757 (Tortosa)

VIC-20

Este juego, al igual que el publicado en el nº 4 correspondiente al mes de mayo "Aventuras en el desierto", tiene que salvarse en cinta, en dos partes por separado.
La primera parte del juego crea los caracteres y mientras muestra las instrucciones se encarga de buscar la segunda parte del juego.

EL JUEGO

Tú conduces un ratón, y tienes que comerte todos los puntos del laberinto, evitando chocar con el gato que te persigue, si te ves acorralado pulsas la letra "G" y te convertirás en un feroz perro. Una vez te hayas comido todos los puntos debes comerte "£", ¡pero no antes!
En la parte superior de la pantalla te indica los ratones y los efectos para convertirte en perro que te quedan. Estos últimos los conseguirás comiendo asteriscos.
Cada tres pantallas te dará un ratón (suponiendo que llegues, claro).

SU DISTRIBUCION

En este juego he omitido las sentencias REM por falta de memoria, pero seguidamente detallaré un poco la distribución del juego y sus principales variables:

10-86 Dibujan el laberinto en pantalla.
90-210 Movimiento del ratón.
215-300 Movimiento del gato.
1000-1020 El gato te ha cogido y un ratón menos.
2000-2070 Comprueba que hayas comido todos los puntos.
3000 Fin del programa y auto run.
4900-5200 Mueven el perro, en caso de haber pulsado "G".
6000-6075 Datas musicales.
8000-8030 Cada tres pantallas te obsequia un ratón.

Principales variables:

Z=Ratones que te quedan.
Y=Efectos para convertirte en perro que te quedan.
D=Número de pantallas.
G=Posición del gato.
Q=Posición del ratón.

Primera Parte

```
10 POKE36879,77:PRINT"[CLR][CRSRD][YEL]
[RVSON][3CRSRR]***CAZA[SPC]GATOS***[3CRSRR]
[RVSOFF]"
30 PRINT"[CRSRD]DEBES[SPC]COMERTE[SPC]
LOS[SPC]PUN-TOS[SPC]Y[SPC]COMO[SPC]U
LTIMO[SPC]´£´"
40 PRINT"[CRSRD]EVITA[SPC]A[SPC]LOS[SPC]
GATOS[SPC]ME-[CRSRR]NOS[SPC]CUANDO[SPC]
SEAS[SPC]PERRO.";
60 PRINTSPC(6)"[2CRSRD]DERECHA[SPC]´
H´":PRINTSPC(5)"[CRSRD]IZQUIERDA[SPC]
´F´":PRINTSPC(6)"[CRSRD]ARRIBA↑[SPC]
´T´"
70 PRINTSPC(7)"[CRSRD]ABAJO[SPC][RVSON]
SPC[RVSOFF]":PRINTSPC(7)"[CRSRD]PERR
O[SPC]´G´"
100 POKE52,28:POKE56,28:CLR
110 FORX=5632TO7679:POKEX,PEEK(X+256
00):NEXTX
120 POKE36869,255
130 FORL=7168TO7175:READD:POKEL,D:NE
XTL
135 FORL=7360TO7383:READD:POKEL,D:NE
XTL
140 DATA34,119,127,107,62,127,54,28
150 DATA255,255,255,255,255,255,255,
255
160 DATA102,231,231,126,90,126,102,2
4
170 DATA102,255,255,90,126,60,255,24

382 PRINT"[CLR][CRSRD][CRSRR]TU[SPC]
LLEVAS[SPC]EL[SPC]RATON[SPC]Z"
384 PRINT"[CRSRD]TE[SPC]SIGUEN[SPC]L
OS[SPC]GATOS[SPC]@":PRINT"[CRSRD][SPC]
CON[SPC]´G´[SPC]ERES[SPC]PERRO[SPC]Y
"
386 PRINT"[CRSRR]PERO[SPC]SOLO[SPC]U
N[SPC]RATITO.[CRSRD][CRSRR]CUANDO[SPC]
COMES[SPC]´*´[SPC]TIE-[CRSRR]NES[SPC]
UN[SPC]EFECTO[SPC]DE[SPC]MAS."
390 PRINT"[CRSRD]CADA[SPC]TRES[SPC]P
ANTALLAS[SPC]TEDARA[SPC]UN[SPC]RATON
.":CLR
400 POKE198,7:POKE631,76:POKE632,207
:POKE633,159:POKE634,13
500 POKE635,82:POKE636,213:POKE637,1
3
```

Segunda Parte

```
1 U=1:Z=3:D=1:POKE36878,15
2 POKE36879,221:FORT=1TO5:PRINT"[BLU]
[CLR][11CRSRD][6SPC]PANTALLA"D:FORO=
1TO300:NEXTO
3 PRINT"[CLR]":FORJ=1TO300:NEXTJ:NEX
TT:POKE36869,255
10 POKE36879,28:PRINT"[CLR][BLK][SPC]
Y=[3SPC]Z=[SPC]":POKE7688,48+Z:PRINT
"[RED]XXXXXXXXXXXXXXXXXXXXXX"
20 PRINT"[CRSRU]X*..................
.X":PRINT"[CRSRU]X.XXX.XXXXXXXXXX.XX
X.X"
25 PRINT"[CRSRU]X...................
.X":PRINT"[CRSRU]X.XXXXXXX.XX.XXXXXX
X.X"
26 PRINT"[CRSRU]X...................
.X":PRINT"[CRSRU]X.XXXXXXXXXXXXXXXXX
X.X"
```

```
40 PRINT"[CRSRU]X...................
.X":PRINT"[CRSRU]X.XXX.X.XXXXXX.X.XX
X.X"
50 PRINT"[CRSRU]X...................
.X":PRINT"[CRSRU]X.XXXXXXXXXXXXXXXXX
X.X"
60 PRINT"[CRSRU]X...................
.X":PRINT"[CRSRU]X.XXXXXXX.XX.XXXXXX
X.X"            "
70 PRINT"[CRSRU]X...................
.X":PRINT"[CRSRU]X.XXX.XXXXXXXXXX.XX
X.X"
80 PRINT"[CRSRU]X*..................
*X":PRINT"[CRSRU]XXXXXXXXXXXXXXXXXXX
XXX[CRSRD]":POKE7683,48+Y
84 IFD>3THENPOKE8033,46
85 IFD>6THENPOKE8052,46
86 IFD>9THENPOKE7725,46
90 G=7735:P=46
100 POKE198,0:M=0:Q=8042:POKEQ,26:PO
KEG,0
110 GETA$:IFA$="T"THENM=1
120 IFA$="[SPC]"THENM=2
130 IFA$="F"THENM=3
140 IFA$="H"THENM=4
145 IFA$="G"THEN4900
150 POKE7734,28:POKEQ,32
160 IFM=1THENQ=Q-V
170 IFM=2THENQ=Q+V
180 IFM=3THENQ=Q-H
190 IFM=4THENQ=Q+H
200 IFPEEK(Q)=24THENV=-V:H=-H:GOTO16
0
202 IFPEEK(Q)=46THENPOKE36877,185:S=
S+1
205 IFQ=GTHEN1000
206 IFQ=7734THEN2000
207 IFPEEK(Q)=42THENY=Y+1:POKE7683,4
8+Y
210 V=22:H=1:POKEQ,26:POKEQ+30720,0:
POKE36877,0:PRINT"[CRSRU]PUNTOS:"S
215 FORT=1TOU:POKEG,P:POKEG+30720,2:
P=46
220 IFG>QTHENN=INT(RND(1)*2)+1
230 IFG<QTHENN=INT(RND(1)*2+3)
240 IFN=1THENG=G-H
250 IFN=2THENG=G-V
260 IFN=3THENG=G+H
270 IFN=4THENG=G+V
280 IFPEEK(G)=24THENH=-H:V=-V:GOTO24
0
285 IFPEEK(G)=32THENP=32
290 POKEG,0:POKEG+30720,6:H=1:V=22:F
ORJ=1TO10:NEXTJ:IFG=QTHEN1000
300 NEXTT
500 GOTO110
1000 POKE36877,0:FORT=15TO0STEP-0.1:
POKE36878,T:POKE36876,200+T:NEXTT:PO
KE36878,10
1010 POKE36876,0:Z=Z-1:IFZ=0THEN3000

1020 GOTO2
2000 POKE36877,0:FORT=7725TO8052:IFP
EEK(T)=46THEN2050
2001 NEXTT
2003 POKE36878,15
2004 RESTORE
2005 READW:IFW=-1THEN2040
2010 READF:POKE36876,W:FORT=1TOF:NEX
TT:GOTO2005
2040 IFU=3THEN8000
2049 U=U+1:D=D+1:POKE36876,0:GOTO2
2050 PRINT"[CRSRD]QUE[SPC]TE[SPC]TIE
NES[SPC]QUE[SPC]CO-[SPC]MER[SPC]TODO
S[SPC]LOS[SPC]PUNTOS!":FORT=1TO5000:
NEXTT
2060 Z=Z-1:IFZ=0THEN3000
2070 GOTO2
3000 PRINT"[CRSRD][6SPC][BLU]GAME[SPC]
OVER.":FORT=1TO5000:NEXTT:RUN
```

```
4900 IFY=0THENGOTO110
5000 Y=Y-1:POKE7683,48+Y:FORT=1TO10:
GETB$:IFB$="T"THENM=1
5010 IFB$="[SPC]"THENM=2
5020 IFB$="F"THENM=3
5030 IFB$="H"THENM=4
5040 POKEQ,32:POKE36877,150+(T*9)
5050 IFM=1THENQ=Q-V
5060 IFM=2THENQ=Q+V
5070 IFM=3THENQ=Q-H
5080 IFM=4THENQ=Q+H
5090 IFPEEK(Q)=24THENV=-V:H=-H:GOTO5
050
5100 V=22:H=1:IFQ=GTHENG=7735:S=S+50

5105 IFPEEK(Q)=46THENS=S+1:PRINT"[CRSRU]
PUNTOS:"S
5107 IFPEEK(Q)=42THENY=Y+1:POKE7683,
48+Y
5110 IFQ=7734GOTO2000
5120 POKEQ,25:POKEQ+30720,0
5130 FORJ=1TO200:NEXTJ
5200 NEXTT:M=0:POKE198,0:GOTO110:POK
E36877,0
6000 DATA195,125,207,250,0,5
6005 DATA207,125,0,5,207,125
6010 DATA201,125,207,125,209,450
6015 DATA207,250,0,5,207,125
6020 DATA201,250,0,5,201,125
6025 DATA0,5,201,125,195,125
6030 DATA201,125,207,500,195,250
6035 DATA0,5,195,125,207,250
6040 DATA0,5,207,125,0,5
6045 DATA207,125,201,125,207,125
6050 DATA209,500,219,750,0,5
6055 DATA219,125,215,250,207,125
6060 DATA209,250,201,125,195,750
6065 DATA215,125,0,5,215,250
6070 DATA0,5,215,125,219,250
6075 DATA0,5,219,125,215,750,-1
8000 D=D+1:U=1:POKE36876,0:POKE36879
,77:PRINT"[CLR][4CRSRD][YEL]*****RE
COMPENSA*****"
8010 PRINT"[2CRSRD][SPC]TE[SPC]OBSEQ
UIO[SPC]UN[SPC]RATON":FORJ=1TO4:A=A+
1:FORT=128TO255:POKE36873+A,T:NEXTT
8020 POKE36873+A,0:NEXTJ:A=A+1:FORJ=
1TO4:A=A-1
8030 FORT=255TO128STEP-1:POKE36873+A
,T:NEXTT:POKE36873+A,0:NEXTJ:A=0:GOT
O2
```



Teclas de función

Para usar las teclas de función en un programa:

```
10 f1$=chr$(133):f2$=chr$(137)
20 f3$=chr$(134):f4$=chr$(138)
   etcétera.
```

Después puedes usar una rutina como...

```
100 geta$:ifa$=""then100
110 ifa$=f1$then xxxx
120 ifa$=f2$then yyyy
130 ifa$=f3$then zzzz
```

Este sistema reduce los errores que cometemos a la hora de depurar programas.

N.S.S. RUN EE.UU.

Simón 64

Francisco Javier Cobano Flores
C/ Don Juan 47
Telf.: (954) 515466 - SEVILLA 41007
Luis Luna Barrios
P/ Gomilla, 16-10
Telf.: (954) 513345 - SEVILLA 41007

C-64



SIMON 64, es un reto a nuestra memoria; se trata de recordar una secuencia de números, sonidos y color, que el ordenador irá presentando progresivamente, secuencia de mayor longitud según el nivel de dificultad elegido. Asimismo no esperará más de treinta segundos su jugada; si esto sucede o se equivoca, tendrá opción a continuar con la misma secuencia o a comenzar desde el principio.

Explicación del programa:

20 a 40	Subrutina de sonido.
230 a 270	Secuencia musical del ordenador.
370 a 400	Toca el ordenador.
410 a 465	Tú tocas.
1600 a 1700	Pinta pantalla.
1700 a 2030	Toca las notas.
5000 a 6000	Explicación.
6000 a 7000	Se ríe de tí.
7000 a 8000	Magnífica presentación.
8000 a 9000	Michael Row the boat ashure-1 Measure.

Sacada del "maravilloso" y "único" manual del usuario.

Además el programa tiene una sorpresita (bastante gansa por cierto) dedicada a los curiosones.

NOTA: 7005 POKE 774,0; si se cambia por un REM, el programa se podrá listar perfectamente, de lo contrario no.

```
1 REM *******************************
**
2 REM *         SIMON 64
 *
3 REM *
 *
4 REM *  COBANO & LUNA   AGOSTO-84
 *
5 REM *
 *
6 REM *
 *
7 REM *    PARA COMMODORE WORLD
 *
8 REM *
 *
```

```
9 REM *********************************
**
10 GOSUB7000:GOTO100
20 FORN=54272TO54295:POKEN,0:NEXT
25 V=54296:POKEV,15
30 POKE54278,240:REM SOS/ESCAPE
35 POKE54276,17:REM ONDA
40 POKE54277,0:REM ATA/CAIDA
45 RETURN
100 PRINT"[CLR]":POKE53280,0:POKE532
81,11:ES$="[HOM][19CRSRD]"
170 PRINT"[CLR][8CRSRD][4SPC]DESEA[SPC]
EXPLICACION[SPC]DEL[SPC]JUEGO(S/N)?"

180 GETA$:IFA$=""THEN180
200 IFA$="S"THENGOSUB5000
210 PRINT"[CLR][8CRSRD][12SPC]NIVEL(
1,2,3)?":Z=54272
220 GETA$:A=VAL(A$):IFA<1ORA>3THEN22
0
230 A1=(1+A)*5:S=0
240 DIMA(A1):REM MATRIZ SECUENCIA OR
D.
250 FORI=1TOA1
260 A(I)=INT(4*RND(0)+1)
270 NEXT
300 REM COLORES PRINCIPIO JUEGO
310 PRINT"[CLR][11CRSRD][9CRSRR]ATEN
CION![SPC]COMENZAMOS"
320 FORT=3TO6:FORG=5TO11STEP2
330 POKE53280,G:POKE53281,T
340 FORK=1TO100:NEXT
350 POKE53280,T:POKE53281,G:NEXT:NEX
T
360 PRINT"[CLR]":GOSUB1600:REM 4 CUA
DRITOS
370 REM SECUENCIA ORD. NOTA A NOTA
375 S=S+1
377 FORT=1TO500:NEXT
380 FORI=1TOS
385 IFI>A1THENGOTO8000:REMGANAS
390 ONA(I)GOSUB1700,1800,1900,2000
400 NEXT
410 J=I:REM NOTAS DEL JUGADOR
420 FORL=1TOI-1
425 TI$="000000"
430 GETB$
432 IFTI$>="000030"THENPRINTES$;"[17SPC]
LENTO!":GOSUB6000:GOTO500
433 IFB$=""THEN430
435 B=VAL(B$):IFB<1ORB>4THEN430
440 IFA(L)<>BTHEN480
450 ONA(L)GOSUB1700,1800,1900,2000
460 NEXT:REM FIN BUCLE JUEGO
465 GOTO375
480 PRINTES$;"[17SPC]FALLO!"
490 GOSUB6000:REM MUSICA FALLO
500 FORT=1TO150:NEXT
510 PRINT"[CRSRU][10SPC]DESEA[SPC]CO
NTINUAR(S/N)"
520 GETA$:IFA$=""THEN520
530 IFA$="S"THEN560
540 IFA$<>"S"ANDA$<>"N"THEN520
550 PRINT"[CLR]":END
560 PRINT"[CLR][3CRSRD][3SPC]PULSE[SPC]
LO[SPC]QUE[SPC]DESEE":PRINT
570 PRINT"[3CRSRD][3SPC]1.[SPC]SEGUI
R[SPC]CON[SPC]LA[SPC]SECUENCIA[SPC]F
ALLADA"
580 PRINT"[3CRSRD][3SPC]2.[SPC]OTRO[SPC]
NIVEL"
590 GETC$:C=VAL(C$):IFC$=""THEN590
600 IFC=1THENS=S-1:GOTO360
610 IFC=2THENCLR:GOTO210
620 GOTO590
1600 REM *********LOS 4 CUADRITOS***
********
1610 PRINT"[CLR]":FORT=1TO5:PRINT:NE
XT
```

```
1620 PRINT"[7SPC][RVSON][RED][11SPC]
[RVSOFF][4SPC][RVSON][WHT][10SPC][RVSOFF]
[COMM7]"
1630 PRINT"[7SPC][RVSON][RED][11SPC]
[RVSOFF][4SPC][RVSON][WHT][10SPC][RVSOFF]
[COMM7]"
1640 PRINT"[7SPC][RVSON][RED][11SPC]
[RVSOFF][4SPC][RVSON][WHT][10SPC][RVSOFF]
[COMM7]"
1650 PRINT"[7SPC][RVSON][RED][7SPC][RVSON]
[YEL]1[RED][3SPC][RVSOFF][4SPC][RVSON]
[WHT][6SPC]2[3SPC][RVSOFF][COMM7]"
1660 PRINT:PRINT:PRINT
1670 PRINT"[7SPC][RVSON][GRN][11SPC]
[RVSOFF][4SPC][RVSON][YEL][10SPC][RVSOFF]
[COMM7]"
1680 PRINT"[7SPC][RVSON][GRN][11SPC]
[RVSOFF][4SPC][RVSON][YEL][10SPC][RVSOFF]
[COMM7]"
1685 PRINT"[7SPC][RVSON][GRN][11SPC]
[RVSOFF][4SPC][RVSON][YEL][10SPC][RVSOFF]
[COMM7]"
1690 PRINT"[7SPC][RVSON][GRN][7SPC]3
[3SPC][RVSOFF][4SPC][RVSON][YEL][6SPC]
4[3SPC][RVSOFF][COMM7]"
1695 RETURN
1697 REM **** LAS CUATRO NOTAS *****
********
1699 REM NOTA 1
1700 GOSUB20:POKE54273,6:POKE54272,2
06
1705 POKE1316,81:POKE1316+Z,0
1710 FORH=1TO250:NEXT
1720 POKE54276,16
1725 POKE1316,160:POKE1316+Z,2
1730 RETURN
1790 REM NOTA 2
1800 GOSUB20:POKE54273,20:POKE54272,
100
1805 POKE1330,81:POKE1330+Z,0
1810 FORH=1TO250:NEXT
1820 POKE54276,16
1825 POKE1330,160:POKE1330+Z,1
1830 RETURN
1890 REM NOTA 3
1900 GOSUB20:POKE54273,81:POKE54272,
161
1905 POKE1596,81:POKE1596+Z,1
1910 FORH=1TO250:NEXT
1920 POKE54276,16
1925 POKE1596,160:POKE1596+Z,5
1930 RETURN
1990 REM NOTA 4
2000 GOSUB20:POKE54273,230:POKE54272
,176
2005 POKE1610,81:POKE1610+Z,1
2010 FORH=1TO250:NEXT
2020 POKE54276,16
2025 POKE1610,160:POKE1610+Z,7
2030 RETURN
5000 REM EXPLICACION
5010 PRINT"[CLR]":PRINTCHR$(14):POKE
646,1
5020 PRINTTAB(13)"[CRSRD][RVSON][SPC]
[SHIFTS][SHIFTI][SHIFTM][SHIFTO][SHIFTN]
[SPC]64[SPC]":PRINT:PRINT
5030 PRINT"[3SPC][SHIFTE]L[SPC][SHIFTS]
[SHIFTI][SHIFTM][SHIFTO][SHIFTN][SPC]
64[SPC]ES[SPC]UN[SPC]JUEGO[SPC]DE[SPC]
MEMORIA."
5040 PRINT"[3SPC][2CRSRD][SHIFTT]RAT
E[SPC]DE[SPC]RECORDAR[SPC]LA[SPC]SEC
UENCIA"
5050 PRINT"[3SPC]DE[SPC]NUMEROS,[SPC]
COLORES[SPC]Y[SPC]SONIDOS"
5060 PRINT"[3SPC]QUE[SPC]EL[SPC]ORDE
NADOR[SPC]LE[SPC]IRA[SPC]OFRECIENDO,
"
5070 PRINT"[3SPC]SUCESIVAMENTE,[SPC]
EN[SPC]ORDEN[SPC]CRECIENTE"
5075 PRINT"[3SPC]SEGUN[SPC]EL[SPC]GR
ADO[SPC]DE[SPC]DIFICULTAD"
5077 PRINT"[3SPC]QUE[SPC]USTED[SPC]P
REVIAMENTE[SPC]ELEGIRA."
5080 PRINT"[3SPC][2CRSRD][SHIFTA]DEM
AS[SPC]NO[SPC]OLVIDE[SPC]QUE[SPC]TIE
NE[SPC]UN"
5090 PRINT"[3SPC]TIEMPO[SPC]MAXIMO[SPC]
DE[SPC]TREINTA[SPC]SEGUNDOS"
5100 PRINT"[3SPC]PARA[SPC]RESPONDER.
"
5110 PRINT"[3SPC][2CRSRD][SHIFTP]ARA
[SPC]CONTINUAR[SPC]PULSE[SPC]UNA[SPC]
TECLA."
5120 GETA$:IFA$=""THEN5120
5130 PRINT"[CLR]":PRINTCHR$(142):POK
E646,14
5140 RETURN
6000 REM MUSICA FALLO
6002 PRINT"[HOM][2CRSRD]"
6005 FORJA=0TO30STEP4
6007 PRINTTAB(JA+4)"JA!":PRINT"[2CRSRU]
"
6010 FORL=54272TO54295:POKEL,0:NEXT
6020 POKE54296,15:REM VOL
6030 POKE54278,255:REM S/E
6040 POKE54276,129:REM ONDA
6050 POKE54277,10:REM A/C
6060 POKE54273,34:POKE54274,75
6065 POKE54276,0:FORT=0TO10:NEXT
6070 NEXT
6080 RETURN
7000 REM PRESENTACION
7005 POKE774,0
7010 PRINT"[CLR]":POKE53280,6:POKE53
281,6
7020 P$="[COMMM][28COMMY][SHIFTM][SPC]
[COMM-][SPC][COMM£][5SPC]"
7030 O$="[COMMM][SPC]COBANO[SPC]&[SPC]
LUNA[4SPC]PRESENTAN:[SPC][SHIFTV][6COMM+]
[COMM£][SPC]"
7040 I$="[COMMM][28COMMP][SHIFTN][4SPC]
[COMM+][4SPC]"
7050 PRINT"[8CRSRD]"
7060 FORX=0TO40
7070 PRINTRIGHT$(P$,X)
7080 PRINTRIGHT$(O$,X)
7090 PRINTRIGHT$(I$,X)
7100 PRINT"[4CRSRU]"
7105 FORT=0TO150:NEXT
7110 NEXT
7120 PRINT"[CLR]"
7130 FORP=0TO10
7140 PRINTTAB(P*2)"[RVSON][SPC]SIMON
[SPC]64[SPC]":PRINT
7150 FORT=0TO100:NEXT:NEXT
7160 FORT=0TO1500:NEXT:RETURN
8000 REM GANAS
8010 PRINT:PRINTTAB(14)"HAS[SPC]GANA
DO":PRINT
8020 PRINTTAB(8)"RECORDASTE";A1;"MOV
IMIENTOS"
8030 V=54296:W=54276:A=54277:HF=5427
3:LF=54272:S=54278:PH=54275:PL=54274
8040 POKEV,15:POKEW,65:POKEA,190:POK
EPH,15:POKEPL,15
8050 READH
8060 READL
8070 READD
8080 IFH=-1THEN 8500
8090 POKEHF,H:POKELF,L
8100 FORX=D-50TOD-20:POKES,136:NEXT
8110 FORT=1TOD:NEXT:POKEHF,0:POKELF,
0:POKEW,0
8120 GOTO8040
8500 PRINT"[HOM][2CRSRD][10SPC]DESEA
[SPC]CONTINUAR[SPC](S/N)"
8510 GETA$:IFA$=""THEN8510
8520 IFA$="S"THENCLR:GOTO210
```

```
8530 IFA$="N"THENPRINT"[CLR]":END
8540 GOTO8510
10000 DATA34,75,250,43,52,250,51,97,
375,43,52,125,51,97,250,51,172,250
10001 DATA51,97,500,0,0,125,43,52,25
0,51,97,250,57,172,1000,51,97,500,-1
,-1,-1
10010 REM        *********
10020 REM        *********
10030 REM        *********
```

Simón

Inmaculada Marquea Aramburo
Mirasierra, 2-4º B
Burgos

VIC-20

En un principio se pide el número de jugadores que van a participar que puede ser desde un solo jugador hasta 10.
Se pedirá el nombre de cada jugador para ser llamados por su nombre cuando corresponda su turno.
A continuación se define el número máximo de notas (a partir de 6 notas la cosa empieza a resultar bastante difícil).
El programa empieza tocando dos notas para cada jugador. Al final de cada ronda se visualizan los fallos de cada uno. En la siguiente ronda se añade una nota más para cada jugador, y así sucesivamente hasta llegar al límite establecido, momento en el que se volverá a empezar con dos notas.
Se dispone de un tiempo determinado para pulsar cada nota, pasado el cual se contabilizará un fallo. Cada vez que se produce un fallo, el ordenador nos muestra cómo era la secuencia correcta de las notas.
Las teclas para cada nota son F, H, B y C como aparece en el dibujo de la pantalla.
Para empezar cada jugador su turno basta con pulsar cualquier tecla.
Este es un buen juego para ejercitar la memoria y los reflejos.

```
80 POKE36879,27
90 PRINT"[CLR][6SPC]S[SPC]I[SPC]M[SPC]
O[SPC]N":J=RND(-TI):POKE36878,15
100 A$(1)="[HOM][5CRSRD][8CRSRR][RVSON]
[SHIFT£][2SPC][CRSRD][4CRSRL][SHIFT£]
[3SPC][CRSRD][5CRSRL][SHIFT£][4SPC][CRSRD]
[6CRSRL][SHIFT£][2SPC]"
105 B$(1)="[RVSON]F[2SPC][CRSRD][6CRSRL]
[6SPC][CRSRD][6CRSRL][6SPC]"
110 A$(2)="[HOM][5CRSRD][11CRSRR][RVSON]
[2SPC][COMM*][CRSRD][3CRSRL][3SPC][COMM*]
[CRSRD][4CRSRL][4SPC][COMM*][CRSRD][5CRSRL]
[2SPC]"
115 B$(2)="[RVSON]H[2SPC][COMM*][CRSRD]
[6CRSRL][6SPC][CRSRD][6CRSRL][6SPC]"

120 A$(3)="[HOM][11CRSRD][11CRSRR][RVSON]
[6SPC][CRSRD][6CRSRL][6SPC][CRSRD][6CRSRL]
[2SPC]"
125 B$(3)="[RVSON]B[2SPC][RVSOFF][SHIFT£]
[RVSON][CRSRD][6CRSRL][4SPC][RVSOFF]
[SHIFT£][RVSON][CRSRD][5CRSRL][3SPC]
[RVSOFF][SHIFT£][RVSON][CRSRD][4CRSRL]
[2SPC][RVSOFF][SHIFT£]"
130 A$(4)="[HOM][11CRSRD][5CRSRR][RVSON]
[6SPC][CRSRD][6CRSRL][6SPC][CRSRD][6CRSRL]
[RVSOFF][COMM*]"+"[RVSON][2SPC]"
131 B$(4)="[RVSON]C[2SPC][CRSRD][5CRSRL]
[RVSOFF][COMM*][RVSON][4SPC][CRSRD][4CRSRL]
[RVSOFF][COMM*][RVSON][3SPC][CRSRD][3CRSRL]
[RVSOFF][COMM*][RVSON][2SPC]"
132 A(1)=225:B(2)=207:B(3)=215:B(4)=
223
133 H$="FHBC"
140 C$="[RED][YEL][BLU][GRN][BLK]":B
(1)=225:B(2)=231:B(3)=235:B(4)=239
145 INPUT"[2CRSRD]LIMITE[SPC]DE[SPC]
NOTAS";L
150 INPUT"[2CRSRD]NUMERO[SPC]DE[SPC]
JUGADORES";J:PRINT"[CLR]"
160 IFJ>10ORJ<1THEN150
165 FORR=1TOJ:PRINT"NOMBRE[SPC]DEL[SPC]
JUGADOR"R
170 INPUTN$(R):NEXT:T=2:POKE36879,23
6:PRINT"[CLR]"
190 FORI=1TO4:PRINTMID$(C$,I,1)A$(I)
B$(I):NEXT
200 FORI=1TOJ:PRINT"[HOM][BLK]JUEGA:
[SPC]"N$(I):PRINT"[HOM][CRSRD]ESCUCH
A.":G$="":G1$=""
201 POKE198,0:WAIT198,1
205 FORZ=1TOT
210 H=INT(RND(1)*4)+1:G1$=G1$+RIGHT$
(STR$(H),1)
220 GOSUB900:G$=G$+MID$(H$,H,1):NEXT
Z
230 FORR=0TO10:GETIN$:NEXT
280 PRINT"[BLK][HOM][CRSRD][7SPC]AHO
RA[SPC]TU"
290 FORZ=1TOT:P=0
300 GETIN$:IFIN$=""THENP=P+1:IFP<500
 THEN300
310 IFIN$<>MID$(G$,Z,1)THEN400
320 H=VAL(MID$(G1$,Z,1)):GOSUB900:NE
XTZ
380 PRINT"[HOM][44SPC]"
385 NEXTI:PRINT"[CLR][BLK]JUGADOR"TA
B(16)"FALLOS"
387 FORW=1TOJ:PRINTN$(W)TAB(16)P(W):
NEXT:POKE198,0
388 WAIT198,1:PRINT"[CLR]"
390 T=T+1:IFT>LTHENT=2
395 GOTO190
400 PRINT"[HOM][44SPC]":POKE36874,20
0:POKE36875,200
410 PRINT"[HOM][BLK][6SPC]INCORRECTO
":FORR=0TO2500:NEXT:PRINT"[HOM][21SPC]
"
420 POKE36874,0:PRINT"[HOM][CRSRD][BLK]
ESCUCHA":POKE36875,0
430 FORZ=1TOT:H=VAL(MID$(G1$,Z,1))
440 GOSUB900:NEXT:P(I)=P(I)+1:GOTO38
5
900 POKE36878,15:POKE36876,A(H):POKE
36875,B(H):PRINT"[BLK]"A$(H)B$(H)
920 FORR=0TO450:NEXT:POKE36876,0:POK
E36875,0
```

```
940 PRINTMID$(C$,H,1)A$(H)B$(H)
950 FORR=0TO10:NEXT:RETURN
```

Cónicas-cuádricas

Francisco Javier Cobano Flores
C/ Don Juan, 47
Telf.: (954) 515466 - SEVILLA 41007

Luis Luna Barrios
P/ Gomila, 16-10
Telf.: (954) 513345 - SEVILLA 41007

C-64

Estos dos programas de aplicación están dentro del campo de la Geometría: clasificación de cónicas y clasificación de cuádricas. Ambos programas, de estructura gemela, pueden librar al estudiante de laboriosos cálculos a la hora de clasificar ambos tipos de curvas, así como para comprobar ejercicios ya hechos. Estos programas están concebidos para, aún a riesgo de ocupar un mayor espacio de memoria con operaciones de cambio de variables, el listado del programa presente las variables con sus nombres clásicos.

Los programas, después de una estética presentación, nos pedirán los datos sobre la ecuación a clasificar, avisándonos de la entrada de datos incorrectos (como puede ser una igualdad 1=0), luego nos dirá el tipo de curva a la que corresponde la ecuación, junto con la misma reducida. En el caso de cuádricas, nos presentará los invariantes.

La estructura de ambos programas no es excesivamente complicada, así que no creemos necesarias más explicaciones.

Cónicas

```
10 REM  @@@@@@@@@@@@@@@@@@@@@@@@@@@@

11 REM  @@@@@@@@@@@@@@@@@@@@@@@@@@@@

12 REM  @                          @

13 REM  @          LUIS            @

14 REM  @                          @

15 REM  @          LUNA            @

16 REM  @                          @

18 REM  @      12 JULIO 1984       @

19 REM  @                          @

20 REM  @@@@@@@@@@@@@@@@@@@@@@@@@@@@

30 REM  @@@@@@@@@@@@@@@@@@@@@@@@@@@@

40 REM
49 PRINT"[CLR]":PRINT"[3CRSRD]"
50 PRINT"[5SPC]**[3SPC]**[2SPC]*[3SPC]
*[2SPC]*[2SPC]**[3SPC]**[3SPC]**"
51 PRINT"[4SPC]*[2SPC]*[SPC]*[2SPC]*
[SPC]*[3SPC]*[2SPC]*[SPC]*[2SPC]*[SPC]
*[2SPC]*[SPC]*[2SPC]*"
52 PRINT"[4SPC]*[4SPC]*[2SPC]*[SPC]*
*[2SPC]*[2SPC]*[SPC]*[4SPC]*[2SPC]*[SPC]
*[3SPC]"
53 PRINT"[4SPC]*[4SPC]*[2SPC]*[SPC]*
*[2SPC]*[2SPC]*[SPC]*[4SPC]*[2SPC]*[SPC]
*[3SPC]"
54 PRINT"[4SPC]*[4SPC]*[2SPC]*[SPC]*
[SPC]*[SPC]*[2SPC]*[SPC]*[4SPC]*[2SPC]
*[SPC]*[3SPC]"
55 PRINT"[4SPC]*[4SPC]*[2SPC]*[SPC]*
[SPC]*[SPC]*[2SPC]*[SPC]*[4SPC]*[2SPC]
*[SPC]*[3SPC]"
56 PRINT"[4SPC]*[4SPC]*[2SPC]*[SPC]*
[SPC]*[SPC]*[2SPC]*[SPC]*[4SPC]****[2SPC]
**[SPC]"
57 PRINT"[4SPC]*[4SPC]*[2SPC]*[SPC]*
[SPC]*[SPC]*[2SPC]*[SPC]*[4SPC]*[2SPC]
*[4SPC]*"
58 PRINT"[4SPC]*[4SPC]*[2SPC]*[SPC]*
[2SPC]**[2SPC]*[SPC]*[4SPC]*[2SPC]*[4SPC]
*"
59 PRINT"[4SPC]*[4SPC]*[2SPC]*[SPC]*
[2SPC]**[2SPC]*[SPC]*[4SPC]*[2SPC]*[4SPC]
*"
60 PRINT"[4SPC]*[4SPC]*[2SPC]*[SPC]*
[3SPC]*[2SPC]*[SPC]*[4SPC]*[2SPC]*[4SPC]
*"
61 PRINT"[4SPC]*[2SPC]*[SPC]*[2SPC]*
[SPC]*[3SPC]*[2SPC]*[SPC]*[2SPC]*[SPC]
*[2SPC]*[SPC]*[2SPC]*"
62 PRINT"[5SPC]**[3SPC]**[2SPC]*[3SPC]
*[2SPC]*[2SPC]**[2SPC]*[2SPC]*[2SPC]
**[SPC]"
63 PRINT"[4CRSRD][15CRSRR][RVSON][SPC]
@[SPC]LUIS[SPC]LUNA[SPC]BARRIOS[SPC]
"
64 FORT=0TO15
65 POKE53280,T:FORT1=0TO500:NEXT
66 NEXT
70 POKE53280,6
100 PRINT"[CLR]":DIMA(3,3)
102 PRINT"[HOM][7CRSRD]"
104 PRINTTAB(11)"[11SPC]2[7SPC]2"
105 PRINTTAB(7)"[SPC][COMM@][2SPC]+[COMM@]
X[SPC]+[COMM@]Y[SPC]+[COMM@]X[SPC]+[COMM@]
XY+[COMM@]Y[SPC]=0"
110 Q1=1024+408
120 FORI=Q1TOQ1+20STEP4
130 POKEI,30:POKEI-4,32
140 INPUT"[CRSRD]";B((I-Q1)/4)
150 NEXT
160 A(0,0)=B(0):A(0,1)=B(1)/2:A(0,2)
=B(2)/2:A(1,1)=B(3):A(1,2)=B(4)/2
161 A(2,2)=B(5)
170 IFB(0)=0ANDB(1)=0ANDB(2)=0ANDB(3
)=0ANDB(4)=0ANDB(5)=0THENGOTO1050
180 IFB(1)=0ANDB(2)=0ANDB(3)=0ANDB(4
)=0ANDB(5)=0THENGOTO1050
200 REM INVARIANTES
205 HH$="[HOM][7CRSRD][4CRSRR]"
210 A00=A(1,1)*A(2,2)-A(1,2)↑2
220 A11=A(0,0)*A(2,2)-A(0,2)↑2
230 A22=A(0,0)*A(1,1)-A(0,1)↑2
240 A=A(0,0)*A(1,1)*A(2,2)+2*A(0,1)*
A(1,2)*A(0,2)
241 B=A(1,1)*A(0,2)↑2+A(0,0)*A(1,2)↑
2+A(2,2)*A(0,1)↑2:DETA=A-B
```

```
300 REM CLASIFICACION
305 PRINT"[CLR]"
310 IFA00=0THEN400
320 IFDETA<>0THEN350
330 IFA00<0THENPRINTHH$;"[RVSON]PAR[SPC]
DE[SPC]RECTAS[SPC]REALES[SPC]SECANTE
S":GOTO1000
340 PRINTHH$;"[RVSON]PAR[SPC]DE[SPC]
RECTAS[SPC]IMAGINARIAS[SPC]SECANTES"
:GOTO1000
350 IFA00<0THENPRINTHH$;"[RVSON]HIPE
RBOLA":GOTO1000
360 Q5=A(1,1)+A(2,2)
365 IFSGN(DETA)=SGN(Q5)THENPRINTHH$;
"[RVSON]ELIPSE[SPC]IMAGINARIA":GOTO1
000
370 PRINTHH$;"[RVSON]ELIPSE[SPC]REAL
":GOTO1000
400 IFDETA<>0THENPRINTHH$;"[RVSON]PA
RABOLA":GOTO500
410 IFA11+A22=0THENPRINTHH$;"[RVSON]
RECTAS[SPC]COINCIDENTES":GOTO550
420 IFA11+A22<0THENPRINTHH$;"[RVSON]
PAR[SPC]DE[SPC]RECTAS[SPC]REALES[SPC]
PARALELAS":GOTO550
430 PRINTHH$;"[RVSON]PAR[SPC]DE[SPC]
RECTAS[SPC]IMAGINARIAS[SPC]PARALELAS
":GOTO550
500 REM ECUACION REDUCIDA DE LA PARA
BOLA
510 PRINTHH$;"[2CRSRD]("; 2*SQR(-DETA
/(A(1,1)+A(2,2)));")X+(";A(1,1)+A(2,
2);")Y↑2=0"
520 GOTO600
550 REM ECUACION REDUCIDA DE R.PARAL
.
552 IF(A(1,1)+A(2,2))=0THEN PRINTHH$
;"[2CRSRD][RVSON]YA[SPC]ESTA[SPC]RED
UCIDA":GOTO600
560 PRINTHH$;"[2CRSRD]("; (A11+A22)/(
A(1,1)+A(2,2));")+(";A(1,1)+A(2,2);"
)Y↑2=0"
600 POKE2020,13:POKE2021,1:POKE2022,
19:POKE2023,63
601 FORWW=56292TO56295:POKEWW,1:NEXT

602 GETR$:IFR$=""THEN601
603 IFR$="S"THENCLR:GOTO100
604 IFR$="N"THENPRINT"[CLR]":POKE532
80,14:END
605 GOTO602
1000 REM REDUCIDA PRIMER TIPO
1010 H=DETA/A00
1020 J=((A(1,1)+A(2,2))-SQR((A(1,1)+
A(2,2))↑2-4*A00))/2
1030 K=((A(1,1)+A(2,2))+SQR((A(1,1)+
A(2,2))↑2-4*A00))/2
1040 PRINTHH$;"[2CRSRD](";H;")+(";K;
")X↑2+(";J;")Y↑2=0":GOTO600
1050 PRINTHH$;"[RVSON][WHT][SPC]DATO
S[SPC]INCORRECTOS[SPC][COMM7]":GOTO6
00
2000 REM    ****************
2001 REM    ****************
2002 REM    ****************
```

Cuádricas

```
10 REM   @@@@@@@@@@@@@@@@@@@@@@@@@@
11 REM   @@@@@@@@@@@@@@@@@@@@@@@@@@
12 REM   @                        @
13 REM   @                        @
14 REM   @   LUIS LUNA BARRIOS    @
15 REM   @                        @
16 REM   @     13 JULIO 1984      @
17 REM   @                        @
18 REM   @                        @
19 REM   @@@@@@@@@@@@@@@@@@@@@@@@@@
20 REM   @@@@@@@@@@@@@@@@@@@@@@@@@@
```

```
40 PRINT"[CLR]"
50 FORG=0TO26STEP2:PRINTTAB(G)"CUADR
ICAS[10SPC][BLK][RVSON][SPC]@[2SPC]L
UIS[SPC]LUNA[COMM7][RVSOFF]"
60 FORH=0TO15:POKE53280,H:NEXT:NEXT
70 FORT=0TO2500:NEXT
99 DIMA(4,4):DIMB(9)
100 PRINT"[CLR]":PRINT"[4CRSRD]":POK
E53280,6
109 PRINTSPC(17)"2"SPC(11)"2"SPC(7)"
2"
110 PRINT"[COMM@][SPC]+[COMM@][SPC]X
+[COMM@][SPC]Y+[COMM@][SPC]Z+[COMM@]
X[SPC]+[COMM@]XY+[COMM@]XZ+[COMM@]Y[SPC]
+[COMM@]YZ+[COMM@]Z[SPC]=0"
120 Q1=1344
130 POKEQ1,30:INPUT"[CRSRD]";B(0):PO
KEQ1,32:Q1=Q1+3
140 FORQ2=Q1TOQ1+32STEP4
150 POKEQ2,30
160 INPUT"";B((Q2-Q1+4)/4)
170 POKEQ2,32
180 NEXT
182 IFB(1)=0ANDB(2)=0ANDB(3)=0ANDB(4
)=0ANDB(5)=0ANDB(6)=0ANDB(7)=0THENL9
=1
184 IFL9=1ANDB(8)=0ANDB(9)=0THEN2100

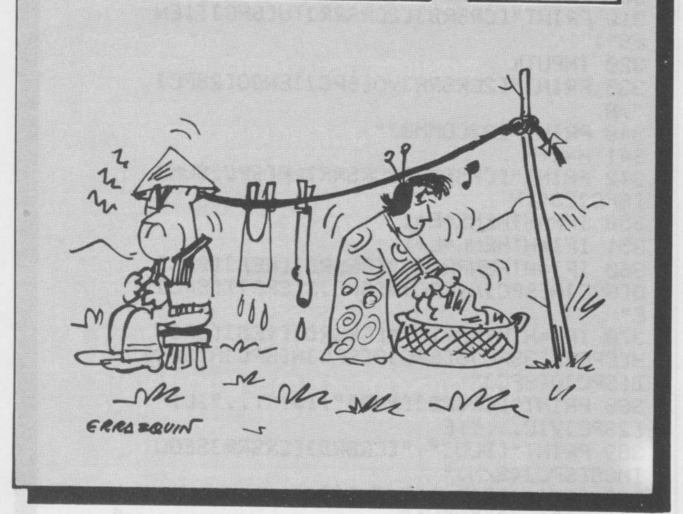
190 A(0,0)=B(0):A(0,1)=B(1)/2:A(0,2)
=B(2)/2:A(0,3)=B(3)/2
192 A(1,1)=B(4):A(1,2)=B(5)/2:A(1,3)
=B(6)/2
194 A(2,2)=B(7):A(2,3)=B(8)/2:A(3,3)
=B(9)
200 REM INVARIANTES
210 REM
220 D1=A(1,1)*A(2,2)*A(3,3)+2*A(1,2)
*A(1,3)*A(2,3)
225 D2=A(2,2)*A(1,3)↑2+A(3,3)*A(1,2)
↑2+A(1,1)*A(2,3)↑2
230 K=D1-D2
240 D3=A(0,1)*A(2,2)*A(3,3)+A(1,2)*A
(2,3)*A(0,3)+A(0,2)*A(2,3)*A(1,3)
250 D4=A(0,3)*A(2,2)*A(1,3)+A(0,2)*A
(1,2)*A(3,3)+A(0,1)*A(2,3)↑2
260 D5=A(0,1)*A(1,2)*A(3,3)+A(1,1)*A
(2,3)*A(0,3)+A(0,2)*A(1,3)↑2
270 D6=A(0,3)*A(1,2)*A(1,3)+A(0,1)*A
(1,3)*A(2,3)+A(0,2)*A(1,1)*A(3,3)
280 D7=A(0,1)*A(1,2)*A(2,3)+A(1,1)*A
(2,2)*A(0,3)+A(0,2)*A(1,3)*A(1,2)
285 D8=A(0,3)*A(1,2)↑2+A(0,2)*A(1,1)
*A(2,3)+A(0,1)*A(1,3)*A(2,2)
290 L=A(0,0)*(D1-D2)-A(0,1)*(D3-D4)+
A(0,2)*(D5-D6)-A(0,3)*(D7-D8)
300 J=(A(1,1)*A(2,2)-A(1,2)↑2)+(A(1,
1)*A(3,3)-A(1,3)↑2)+(A(2,2)*A(3,3)-A
(2,3)↑2)
310 I=A(1,1)+A(2,2)+A(3,3)
320 C1=A(0,0)*A(2,2)*A(3,3)+2*A(0,2)
*A(2,3)*A(0,3)
325 C2=A(2,2)*A(0,3)↑2+A(0,0)*A(2,3)
↑2+A(3,3)*A(0,2)↑2
330 C3=A(0,0)*A(1,1)*A(3,3)+2*A(0,1)
*A(1,3)*A(0,3)
335 C4=A(1,1)*A(0,3)↑2+A(3,3)*A(0,1)
↑2+A(0,0)*A(1,3)↑2
340 C5=A(0,0)*A(1,1)*A(2,2)+2*A(0,1)
*A(1,2)*A(0,2)
345 C6=A(1,1)*A(0,2)↑2+A(0,0)*A(1,2)
↑2+A(2,2)*A(0,1)↑2
350 KT=C1-C2+C3-C4+C5-C6
360 JT=(A(0,0)*A(1,1)-A(0,1)↑2)+(A(0
,0)*A(2,2)-A(0,2)↑2)+(B(0)*A(3,3)-A(
0,3)↑2)
370 REM SIGNATURA
380 W1=SGN(K):W2=SGN(J):W3=SGN(I):W4
=SGN(1)
390 IFW1=0THENW1=1
400 IFW2=0THENW2=1
```

```
410 IFW3=0THENW3=1
420 IFW1=W2THENPE=PE+1
430 IFW1<>W2THENVA=VA+1
440 IFW2=W3THENPE=PE+1
450 IFW2<>W3THENVA=VA+1
460 IFW3=W4THENPE=PE+1
470 IFW3<>W4THENVA=VA+1
480 SIG=ABS(PE-VA)
500 REM CLASIFICACION
505 PH$="[HOM][7CRSRD][2CRSRR]"
510 HH$="[CLR][5CRSRD][2CRSRR][RVSON]
"
520 IFL=0THEN750
530 IFK<>0THEN570
540 IFJ>0THENPRINTHH$;"PARABOLOIDE[SPC]
ELIPTICO":GOTO1000
550 IFJ<0THENPRINTHH$;"PARABOLOIDE[SPC]
HIPERBOLICO":GOTO1000
570 IFSIG=1THEN600
580 IFL>0THENPRINTHH$;"ELIPSOIDE[SPC]
IMAGINARIO":GOTO1400
590 IFL<0THENPRINTHH$;"ELIPSOIDE[SPC]
REAL":GOTO1400
600 IFL>0THENPRINTHH$;"HIPERBOLOIDE[SPC]
HIPERBOLICO":GOTO1400
610 IFL<0THENPRINTHH$;"HIPERBOLOIDE[SPC]
ELIPTICO":GOTO1400
750 REM L=0
760 IFK=0THEN780
765 IFSIG=1THENPRINTHH$;"CONO[SPC]RE
AL":GOTO1400
775 PRINTHH$;"CONO[SPC]IMAGINARIO":G
OTO1400
780 IFJ=0THEN850
790 IFJ>0THEN820
800 IFKT<>0THENPRINTHH$;"CILINDRO[SPC]
HIPERBOLICO":GOTO1100
810 IFKT=0THENPRINTHH$;"PLANOS[SPC]R
EALES[SPC]SECANTES":GOTO1100
820 IFKT=0THENPRINTHH$;"PLANOS[SPC]I
MAGINARIOS[SPC]SECANTES":GOTO1100
830 IFSGN(KT)=SGN(I)THENPRINTHH$;"CI
LINDRO[SPC]IMAGINARIO":GOTO1100
840 PRINTHH$;"CILINDRO[SPC]ELIPTICO[SPC]
REAL":GOTO1100
850 IFI=0THENPRINTHH$;"PLANO[SPC]UNI
CO":PRINT"[CRSRD][RVSON][2CRSRR]YA[SPC]
ESTA[SPC]REDUCIDO":GOTO2000
860 IFKT<>0THENPRINTHH$;"CILINDRO[SPC]
PARABOLICO":GOTO1200
870 IFJT>0THENPRINTHH$;"PLANOS[SPC]I
MAGINARIOS[SPC]PARALELOS":GOTO1300
880 IFJT<0THENPRINTHH$;"PLANOS[SPC]R
EALES[SPC]PARALELOS":GOTO1300
890 PRINTHH$;"PLANOS[SPC]COINCIDENTE
S":GOTO1300
899 REM
900 REM ECUACIONES REDUCIDAS
920 REM
1000 REM PARABOLOIDE
1005 IF(I↑2-4*J)<0THEN2100
1006 IF-L/J<0THEN2100
1010 K1=SQR(I↑2-4*J)
1020 PRINTPH$
1030 PRINT"(";(I+K1)/2;")X↑2+(";(I-K
1)/2;")Y↑2+(";2*SQR(-L/J);")Z=0":GOT
O2000
1100 REM CILINDROS NO PARABOLICOS
         PLANOS SECANTES
1110 IF(I↑2-4*J)<0THEN2100
1120 K1=SQR(I↑2-4*J)
1130 PRINTPH$
1140 PRINT"(";KT/J;")+(";(I-K1)/2;")
X↑2+(";(I+K1)/2;")Y↑2=0":GOTO2000
1200 REM CILINDRO PARABOLICO
1209 IF(-KT/I)<0THEN2100
1210 PRINTPH$
1220 PRINT"(";I;")X↑2+(";2*SQR(-KT/I
);")Y=0":GOTO2000
1300 REM PLANOS PARALELOS
1310 PRINTPH$
1320 PRINT"(";JT/I;")+(";I;")X↑2=0":
GOTO2000
1400 REM ELIPSOIDES HIPERBOLOIDES CO
NOS
1410 PRINTPH$;"(";L/K;")+(A1)X↑2+(A2
)Y↑2+(A3)Z↑2=0"
1420 PRINT:PRINT"CON[SPC]A1,[SPC]A2,
[SPC]A3,[SPC]LAS[SPC]SOLUCIONES[SPC]
DE:"
1430 PRINT"[3SPC]P↑3-(";I;")P↑2+(";J
;")P-(";K;")=0"
1440 GOTO2000
2000 POKE2020,13:POKE2021,1:POKE2022
,19:POKE2023,63
2010 FORKK=56295TO56295-4STEP-1:POKE
KK,1:NEXT
2014 IFLW=1THEN2020
2015 PRINT"[4CRSRD][2CRSRR]L=";L;"*K
=";K;"*J=";J;"*I=";I;"*SIG=";SIG
2017 PRINT"[2SPC]KT=";KT;"*JT=";JT
2020 GETEA$:IFEA$=""THEN2020
2030 IFEA$="S"THENCLR:GOTO100
2040 IFEA$<>"N"THEN2020
2050 POKE53280,14:PRINT"[CLR]":END
2100 PRINTTAB(15)"[WHT][RVSON][SPC]D
ATOS[SPC]INCORRECTOS[SPC][COMM7]":LW
=1:GOTO2000
2200 REM      ****************
2201 REM      ****************
2202 REM      ****************
```



Julián Alvarez García
Fernández de la Puente, 11-9º B
Badajoz

VIC-20
sin ampliación



Es el clásico juego de los chinos. Es tan sencillo que pienso que no es necesaria explicación.

```
1 PRINT"[CLR][20CRSRD][RED]COPYGRIGH
T[SPC]JULIAN[BLU]"
3 FORD=1TO7:PRINT"[CRSRD]":NEXT
4 PRINT"COMO[SPC]TE[SPC]LLAMAS",
```

```
5 INPUTI$
10 PRINT"[CLR]"
20 POKE36879,255
30 A=INT(RND(1)*3+1)
40 PRINT "[2CRSRR]TU[SPC]PIDES",
50 INPUTI
60 X=INT(I/2)+A
70 IFX=ITHENX=X-1
72 IF I=0THEN X=A
73 IFI=6 THENX=A+3
80 PRINT"[CRSRD][2CRSRR]YO[SPC]PIDO[3SPC]
";X
85 PRINT"[22COMM@][CRSRD]"
90 PRINT"[2CRSRR]LLEVAS",
100 INPUTK
102 PRINT"[CRSRD][2CRSRR]LLEVO[5SPC]
";A
103 PRINT "[22COMM@]"
110 PRINT"[2CRSRR]LA[SPC]SUMA[SPC]ES
";
120 U=A+K
130 PRINTU
170 IFX=UTHENPRINT"[CRSRD][2CRSRR][RED]
H[SPC]E[3SPC]G[SPC]A[SPC]N[SPC]A[SPC]
D[SPC]O"
180 IFI=UTHENPRINT"[CRSRD][2CRSRR][RED]
G[SPC]A[SPC]N[SPC]A[SPC]S[SPC]T[SPC]
E"
181 IFI=UTHENL=L+1
182 IFX=UTHENE=E+1
183 PRINT"[CRSRD][PUR]";I$;"...";L"[2SPC]
VIC...";E
200 PRINT"[BLU][RVSOFF]"
210 PRINT"[CRSRU][CRSRR][CRSRD]SEGUI
MOS[SPC](S/N)?"
220 GETG$:IFG$<>"S"THEN220
230 PRINT"[CLR]"
240 A=INT(RND(1)*4)
250 P=INT(RND(1)*7)
260 IFP<ATHEN240
261 IFP>A+3THEN250
270 PRINT"[CRSRD][2CRSRR]YO[SPC]PIDO
[3SPC]";P
280 PRINT"[CRSRD][2CRSRR]TU[SPC]PIDE
S",
290 INPUTI
300 PRINT"[22COMM@]"
310 PRINT"[CRSRD][2CRSRR]TU[SPC]TIEN
ES";
320 INPUTK
330 PRINT"[2CRSRR]YO[SPC]TENGO[2SPC]
";A
340 PRINT"[22COMM@]"
341 H=K+A
342 PRINT"[CRSRD][2CRSRR]LA[SPC]SUMA
[SPC]ES";H
350 IFP=HTHENE=E+1
351 IFI=HTHENL=L+1
360 IFI=HTHENPRINT"[CRSRD][RED][SPC]
G[SPC]A[SPC]N[SPC]A[SPC]S[SPC]T[SPC]
E"
370 IFP=HTHENPRINT"[CRSRD][RED][SPC]
H[SPC]E[3SPC]G[SPC]A[SPC]N[SPC]A[SPC]
D[SPC]O[SPC]"
388 PRINT"[CRSRD][PUR]";I$;"...";L;"
[2SPC]VIC...";E
389 PRINT"[BLU]";"[CRSRD][CRSRR]SEGU
IMOS[SPC](S/N)"
390 GETG$:IFG$=""THEN390
391 IFG$="S"THEN10
392 IFG$="N"THENPRINT"[CLR]":END
393 GOTO390
399 PRINT"[BLU]"
400 GOTO10
```

Reloj de mesilla

José Miguel Suñe Ortín
Pza. de Aragón, 7-6º Izqda.
50004 Zaragoza

C-64



Al ejecutar el programa, el ordenador pide una serie de datos: hora, fecha, hora de la alarma y sonido de la misma. La hora hay que entrarla de la forma habitual (ej. 032648). Este número ha de ser siempre menor a 125959. Si, por ejemplo, quisiéramos introducir las 7 h. de la tarde, tendríamos que poner 070000, y contestar a la siguiente pregunta AM(1), PM(2), con un 2.

Los sonidos de la alarma los he sacado de las págs. 162 y 153-154 del manual del usuario. Cada uno puede cambiarlos a voluntad (líneas últimas).

Después de haber contestado a todas las preguntas, el ordenador pasa a dibujar el reloj; en primer lugar, y hay que apretar la tecla F7, para que la alarma pueda sonar (igual que un reloj de mesilla).

Apretando F1, aparecerá el día y el mes; y pulsando otra vez F7 al sonar la alarma, parará la ejecución de la misma. Si la alarma no se para con este procedimiento, sonará un minuto (más o menos).

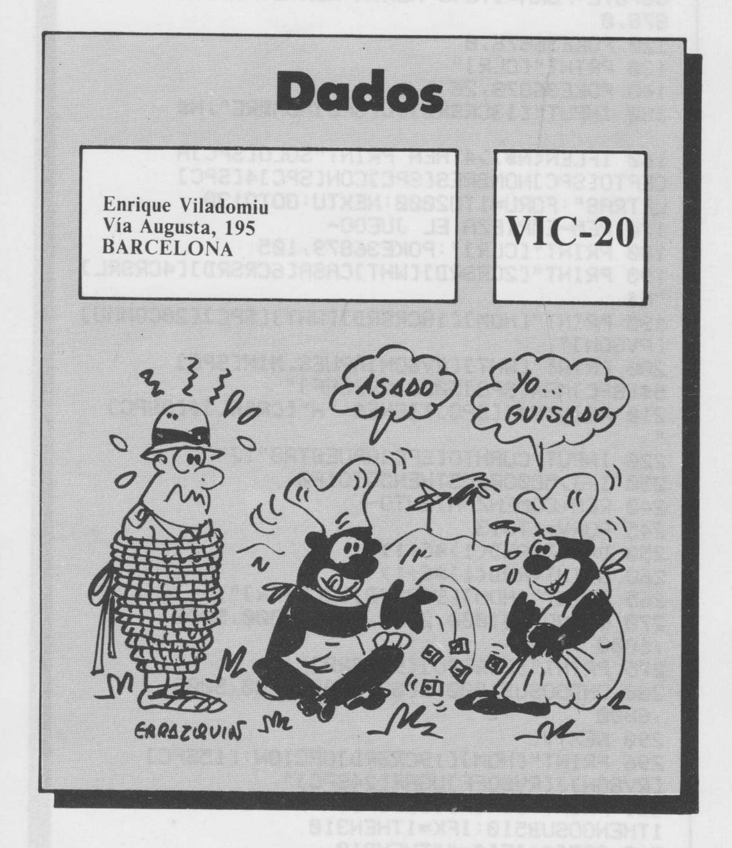
```
0 REM * RELOJ DE MESILLA *
5 REM * COPYRIGHT (C) J. MIGUEL SUNE
 - VERANO 1984 *
10 PRINTCHR$(142):POKE53280,1:POKE53
281,1
15 FORI=54272TO54278:POKEI,0:NEXTI:P
OKE54296,0
20 INPUT"[CLR][CRSRD][BLK][SPC]INTRO
DUZCA[SPC]LA[SPC]HORA([SPC]MENOR[SPC]
QUE[SPC]125959[SPC])[SPC]";TI$
25 IFTI$>"125959"THENRUN
30 INPUT"[CRSRD][SPC]AM[SPC](1)[SPC]
O[SPC]PM[SPC](2)[SPC]";J
35 INPUT"[CRSRD][SPC]MES[SPC]:[SPC]"
;A
40 INPUT"[CRSRD][SPC]DIA[SPC]:[SPC]"
;F
45 INPUT"[CRSRD][SPC]HORA[SPC]DE[SPC]
LA[SPC]ALARMA[SPC]:[SPC]";TI$(R):INP
UT"[CRSRD][SPC]AM[SPC](1)[SPC]O[SPC]
PM[SPC](2)[SPC]";H
50 PRINT"[CRSRD][SPC]SONIDO[SPC]DE[SPC]
LA[SPC]ALARMA[SPC]:":PRINT"[CRSRD][SPC]
```

```
G=GRAVE","M=MEDIO","A=AGUDO"
55 INPUT"[SPC]CUAL[SPC]";H$:IFH$<>"A
"ANDH$<>"M"ANDH$<>"G"THEN50
60 GOSUB275
65 PRINT"[CLR][GRN][6CRSRR][RVSON][3SPC]
RELOJ[4SPC]DE[4SPC]MESILLA[3SPC]"
70 PRINT:PRINT"[3CRSRD][3CRSRR][RVSON]
[34SPC][RVSOFF][SPC]"
75 PRINT"[GRN][3CRSRR][RVSON][SPC][RVSOFF]
[32SPC][RVSON][SPC]"
80 PRINT"[3CRSRR][RVSON][SPC][RVSOFF]
[32SPC][RVSON][SPC][BLU]F[GRN]"
85 PRINT"[3CRSRR][RVSON][SPC][RVSOFF]
[32SPC][RVSON][SPC][BLU]1[GRN]"
90 PRINT"[3CRSRR][RVSON][SPC][RVSOFF]
[11SPC][COMMA][9SHIFT*][COMMS][10SPC]
[RVSON][SPC]"
95 PRINT"[3CRSRR][RVSON][SPC][RVSOFF]
[11SPC][SHIFTB][9SPC][SHIFTB][10SPC]
[RVSON][SPC]"
100 PRINT"[3CRSRR][RVSON][SPC][RVSOFF]
[11SPC][SHIFTB][9SPC][SHIFTB][10SPC]
[RVSON][SPC]"
105 PRINT"[3CRSRR][RVSON][SPC][RVSOFF]
[11SPC][SHIFTB][9SPC][SHIFTB][10SPC]
[RVSON][SPC]"
110 PRINT"[3CRSRR][RVSON][SPC][RVSOFF]
[11SPC][COMMZ][9SHIFT*][COMMX][10SPC]
[RVSON][SPC][BLU]F[GRN]"
115 PRINT"[3CRSRR][RVSON][SPC][RVSOFF]
[32SPC][RVSON][SPC][BLU]7[GRN]"
120 PRINT"[3CRSRR][RVSON][SPC][RVSOFF]
[32SPC][RVSON][SPC]"
125 PRINT"[3CRSRR][RVSON][34SPC][RVSOFF]
"
130 PRINT"[4CRSRR][SHIFTQ][30SPC][SHIFTQ]
[4SPC]"
135 PRINT"[HOM][9CRSRD][6CRSRR]"LEFT
$(TI$(R),2)"[SPC]-[SPC]"MID$(TI$(R),
3,2)
140 IFH=1THENPRINT"[8CRSRR]-AM-"
145 IFH>1THENPRINT"[8CRSRR]-PM-"
150 PRINT"[BLK][HOM][10CRSRD][10CRSRL]
[SPC]:";RIGHT$(TI$,2)
155 IFTI$>"125959"THENTI$="010000":J
=J+1
160 IFJ=1THENM$="AM":GOSUB270
165 IFJ>1THENM$="PM":GOSUB270:J=0
170 PRINT"[BLK][HOM][11CRSRD][9CRSRR]
[6SPC][2CRSRR]"LEFT$(TI$,2);"[SPC]:[SPC]
";MID$(TI$,3,2)
175 IFTI$=TI$(R)AND L=1THENGOSUB290
180 GETA$:IFA$="[F1]"THENGOSUB215:GO
SUB200:GOTO150
185 IFA$="[F7]"THENL=L+1:IFL=1THENGO
SUB230:GOTO150
190 IFL>1THENGOSUB245:L=0:GOTO150
195 GOTO135
200 PRINT"[HOM][6CRSRD][5CRSRR]MES[SPC]
:[2SPC]";A;"[12CRSRR]DIA[SPC]:"F
205 FORI=1TO90:GOSUB320:NEXT
210 PRINT"[HOM][6CRSRD][4CRSRR][31SPC]
[4CRSRR]":GOSUB260:RETURN
215 PRINT"[HOM][6CRSRD]":FORB=1TO2:P
RINT"[CRSRD][3CRSRL][SPC]"
220 NEXT
225 RETURN
230 PRINT"[HOM][4CRSRD][7CRSRR][RVSON]
[BLU][SPC]F-7[SPC][RVSOFF]":L=1
235 PRINT"[9CRSRD][3CRSRL][3SPC]"
240 PRINT"[3CRSRL][3SPC]":RETURN
245 PRINT"[HOM][4CRSRD][7CRSRR][RVSOFF]
[14SPC]":L=0
250 PRINT"[9CRSRD][3CRSRL][RVSON][BLU]
F"
255 PRINT"[3CRSRL][RVSON][BLU]7[GRN]
":RETURN
260 PRINT"[HOM][7CRSRD][2SPC][2CRSRR]
[32SPC][CRSRR][RVSON][BLU]F"
265 PRINT"[2SPC][2CRSRR][32SPC][CRSRR]
[RVSON]1[GRN][5CRSRR]":RETURN
270 PRINT"[HOM][12CRSRD][13CRSRL]";M
$:RETURN
275 IFH$="A"THENV=96:K=254:RETURN:RE
M*F#-6
280 IFH$="M"THENV=34:K=75:RETURN:REM
*C-5
285 IFH$="G"THENV=11:K=114:RETURN:RE
M*F-3
290 POKE54296,15:POKE54277,2:POKE542
78,240:POKE54276,33:FORI=1TO28
295 POKE54272,K:POKE54273,V:FORZ=1TO
700:NEXTZ:POKE54296,0:GOSUB320:FORY=
1TO500
300 NEXTY:GOSUB320:POKE54296,15
305 GOSUB320:GETA$:IFA$="[F7]"THENL=
2:GOTO315
310 NEXTI
315 POKE54296,0:POKE54272,0:POKE5427
3,0:L=2:RETURN
320 PRINT"[HOM][10CRSRD][8CRSRL]"RIG
HT$(TI$,2):RETURN
```

Dados

Enrique Viladomiu
Vía Augusta, 195
BARCELONA

VIC-20



La explicación del programa es muy sencilla. Antes de todo te pide tu nombre, después imprime tu nombre en la pantalla junto al de tu contrincante, la casa, que en este caso es la computadora. Después te pide la cantidad que quieres apostar de 5 a 500$. Cuando lo hayas escrito aparecerán los datos en la pantalla moviéndose y cambiando los números hasta finalmente parar. En la primera jugada la única opción es la de jugar. Después de la primera jugada ya te pide la opción de jugar o de cubrirse. Cuando eliges jugar la casa muestra sus dados. Si tus dados son más altos que los de la casa, ganas lo que has apostado, si tu puntuación es más baja, pierdes lo que habías apostado. Si la opción es cubrir, pierdes la mitad de lo que has apostado, te enseña en la parte inferior de la pantalla lo que hubiese tenido la casa y pasa seguidamente a la próxima jugada. Lo ganado se va almacenando y te lo va mostrando cuando apuestas.

```
4 PRINT"[CLR]"
5 POKE36879,26
7 H=1000:K=0
10 REM-JUEGO DE DADOS-
15 REM--PRESENTACION--
16 PRINT"[CLR]"
17 FOR A=1TO22
20 PRINT"[SHIFTQ]";
25 NEXTA
27 PRINT"[17CRSRD]"
30 FORA=1TO22
35 PRINT"[SHIFTQ]";
40 NEXTA
50 FORB=1TO500:NEXT
60 PRINT"[HOM][10CRSRD][8CRSRR][RVSON]
DADOS[RVSOFF](C)"
70 PRINT"[2CRSRD][20CRSRL]ENRIQUE[SPC]
VILADOMIU"
80 PRINT"[2CRSRD][16CRSRL]BARCELONA"

81 FORA=0TO2000:NEXTA
90 PRINT"[8CRSRD][19CRSRL]PULSA[SPC]
UNA[SPC]TECLA"
100 GETA$:IFA$=""THEN100
110 POKE36878,15:FORL=130TO254:POKE3
6876,L:FORM=1TO40:NEXTM:NEXTL:POKE36
878,0
120 POKE36876,0
130 PRINT"[CLR]"
140 POKE36879,26
150 INPUT"[13CRSRD]TU[SPC]NOMBRE";N$

160 IFLEN(N$)>4THEN PRINT"SOLO[SPC]A
CEPTO[SPC]NOMBRES[SPC]CON[SPC]4[SPC]
LETRAS":FORU=1TO2000:NEXTU:GOTO130
170 REM-EMPIEZA EL JUEGO-
180 PRINT"[CLR]":POKE36879,105
190 PRINT"[2CRSRD][WHT]CASA[6CRSRD][4CRSRL]
"N$
195 PRINT"[HOM][18CRSRD][WHT][SPC][20COMMU]
[RVSON]"
200 PRINT"[WHT][RVSON]APUES.MIN[SPC]
5$[SPC]MAX[SPC]500$[RVSOFF]"
210 PRINTN$"[SPC]TIENES:"H"[CRSRL]$[3SPC]
"
220 INPUT"CUANTO[SPC]APUESTAS";J
230 IFJ>500ORJ<5THENGOTO180
240 REM-COMENZAMIENTO-
245 FORV=1TO15
250 D=INT(RND(1)*6+1)
260 S=INT(RND(1)*6+1)
265 PRINT"[HOM][9CRSRD][3CRSRR]";
270 ONSGOSUB1000,2000,3000,4000,5000
,6000
275 PRINT"[3CRSRU][9CRSRR]";
280 ONDGOSUB1000,2000,3000,4000,5000
,6000
290 NEXTV
296 PRINT"[HOM][19CRSRD]OPCION:[15SPC]
[RVSON]J[RVSOFF]UGAR[24SPC]"
301 PRINT"[CRSRU][21SPC]"K=K+1:IFK>
1THENGOSUB510:IFK=1THEN310
310 GETI$:IFI$=""THEN310
311 IFI$="J"THENGOTO400
320 IFI$="C"THENPRINT"[HOM][10CRSRD]
*PI"INT(J/2):H=H-INT(J/2):GOTO996
400 FORV=1TO15
410 W=INT(RND(1)*6+1)
420 X=INT(RND(1)*6+1)
425 PRINT"[HOM][3CRSRD][3CRSRR]";
430 ON WGOSUB1000,2000,3000,4000,500
0,6000
435 PRINT"[HOM][3CRSRD][9CRSRR]";
440 ON XGOSUB1000,2000,3000,4000,500
0,6000
445 NEXTV
446 IFD+S=W+XTHENGOTO449
447 IFD+S<W+XTHENGOTO450
448 IFD+S>W+XTHENGOTO460
449 IFD+S=W+XTHENPRINT"[HOM][10CRSRD]
*IGUAL":GOTO997
450 IFD+S<W+XTHENPRINT"[HOM][10CRSRD]
*PI"J:FORA=1TO1000:NEXTA:H=H-J
451 POKE36877,220:FORL=15TO0STEP-1:P
OKE36878,L:FORM=1TO300:NEXT:NEXT:POK
E36877,0
452 POKE36878,0:GOTO996
460 IFD+S>W+XTHENPRINT"[HOM][10CRSRD]
*GAN"J:FORA=1TO1000:NEXTA:H=H+J
461 FORA=1TO5:POKE36878,15:FORL=130T
O254:POKE36876,L:NEXTL: POKE36878,0:
POKE36876,0:NEXT
462 GOTO996
510 PRINT"[HOM][19CRSRD][5CRSRR][CRSRD]
/[RVSON]C[RVSOFF]UBRIR":RETURN
996 PRINT"[HOM][20CRSRD][21SPC]"
997 PRINT"[HOM][19CRSRD][RVSON]"N$"=
"D+S"[CRSRL][5SPC]CASA="W+X:FORA=1TO
3000:NEXT:GOTO170
998 GETA$:IFA$="S"THEN170
1000 PRINT"[6CRSRR][RVSON][3SPC][CRSRD]
[3CRSRL][SPC][SHIFTQ][SPC][CRSRD][3CRSRL]
[3SPC][RVSOFF]":RETURN
2000 PRINT"[6CRSRR][RVSON][SHIFTQ][2SPC]
[CRSRD][3CRSRL][3SPC][CRSRD][3CRSRL]
[2SPC][SHIFTQ][RVSOFF]":RETURN
3000 PRINT"[6CRSRR][RVSON][SHIFTQ][2SPC]
[CRSRD][3CRSRL][SPC][SHIFTQ][SPC][CRSRD]
[3CRSRL][2SPC][SHIFTQ][RVSOFF]":RETU
RN
4000 PRINT"[6CRSRR][RVSON][SHIFTQ][SPC]
[SHIFTQ][CRSRD][3CRSRL][3SPC][CRSRD]
[3CRSRL][SHIFTQ][SPC][SHIFTQ][RVSOFF]
":RETURN
5000 PRINT"[6CRSRR][RVSON][SHIFTQ][SPC]
[SHIFTQ][CRSRD][3CRSRL][SPC][SHIFTQ]
[SPC][CRSRD][3CRSRL][SHIFTQ][SPC][SHIFTQ]
[RVSOFF]":RETURN
6000 PRINT"[6CRSRR][RVSON][SHIFTQ][SPC]
[SHIFTQ][CRSRD][3CRSRL][SHIFTQ][SPC]
[SHIFTQ][CRSRD][3CRSRL][SHIFTQ][SPC]
[SHIFTQ][RVSOFF]":RETURN
```



Nombres de colores

Si quieres poner una nota de color en los nombres de tus programas haz lo siguiente:

Grábalo con el nombre:

"[CTRL 1]N[CTRL 2]O[CTRL 3]M[CTRL 4]B[CTRL 5]R[CTRL 6[E"

Ten en cuenta que cada control de color ocupa un carácter de los 16 permitidos.

J.S.P. RUN EE.UU.

Load/Run

Todos sabemos que podemos cargar y autoejecutar un programa desde cinta pulsando SHIFT y RUN/STOP.

El problema es que no se puede hacer lo mismo con los programas en disco. Para solucionarlo, se puede hacer:

LOAD"PROGRAMA",8 y pulsar SHIFT-Return.

Esto hará que NO se ejecute la línea anterior.

Entonces subimos con el cursor y pulsamos SHIFT-RUN/STOP. Ahora el programa se cargará y se autoejecutará.

M.F. RUN EE.UU.

Bit Map Inmediato

(Envíanos tus señas)

C-64

Os envío este programa animado por lo que cuenta Ignacio Ortega en el número 8 de la revista, a mí me paso lo mismo que él cuenta con respecto al manual del usuario y después con la guía de referencia. Tengo el C-64 y el cartucho HES MON 64 desde julio, cuando empece a ver la guía de referencia me llamó mucho la atención el modo "Bit Map" de alta resolución, pero al probar el programa ejemplo de la guía me desilusione un poco al ver la lentitud del método (llenado de la pantalla de "porquería", posterior borrado línea a línea, y finalmente el cambio de color), así pues, que me decidí a hacer una subrutina en código máquina, ayudado por el ensamblador del Hes Mon.

El programa yo lo tengo en casa con todas las funciones que se me han ido ocurriendo con el tiempo, como me faltan los números 1, 3, 4 y 5 de la revista siempre pensé que ya lo habríais publicado.

Aquí os lo mando con tres funciones (Coseno, elipse y epicicloide) para que podáis ver cómo se usa la subrutina y para ayudar un poco al que no entienda aún bien la manera en que se han de expresar las fórmulas de las funciones a representar. También he introducido la esfera para que ayudados de vuestra imaginación y sabiduría matemática podáis representar figuras aparentemente en tres dimensiones.

El programa comienza en la línea 20, para que una vez efectuado el primer Run, añadáis la línea 10 que debe ser:

10 GOTO 30

Esto es porque al efectuar el primer RUN, por medio de las instrucciones READ y DATA se introduce en memoria a partir de la posición 49.152 ($C000) la subrutina en código máquina. En esta la subrutina queda salvada del programa en BASIC, es decir que no corre peligro de borrarse, así pues, la primera vez que se ejecuta el programa será la más lenta.

Líneas 30-90. Estas líneas no creo que necesiten explicación, simplemente sirven para elegir la función que deseas ver representada. Podéis cambiarlas o añadir las que vosotros queráis.

Líneas 100-130. Con estas líneas se representa la función COS, las funciones CH, RO, LN, BY y BI son las que explica la Guía de Referencia.

Líneas 140-210. Con estas se representa una elipse en cualquier parte de la pantalla y del tamaño que vosotros elijáis.

Líneas 220-330. Por medio de estas se representa en la pantalla una epicicloide de la cual vosotros debéis elegir los radios de la circunferencia fija y de la circunferencia generatriz móvil. Se vera representada en el medio de la pantalla, además al final también se representa la circunferencia fija.

Líneas 340-470. Con estas se representa una esfera en el centro de la pantalla.

Línea 990. Es la forma que explica la Guía de introducir una subrutina en lenguaje máquina a partir de BASIC.

```
20 GOSUB990
30 PRINT"[CLR]""[RVSON][WHT]DIME[SPC]
LA[SPC]FUNCION[SPC]QUE[SPC]QUIERES[SPC]
VER[SPC]REPRESENTADA"
40 PRINT"COS,ELIPSE,EPICICLOIDE,ESFE
RA"
50 INPUTA$:IFA$=""THENGOTO20
60 IFA$="COS"THENGOTO100
70 IFA$="ELIPSE"THENGOTO140
80 IFA$="EPICICLOIDE"THENGOTO220
90 IFA$="ESFERA"THENGOTO340
100 SYS49152
110 FORX=0TO300:Y=100-90*COS(X*π/150
):CH=INT(X/8):RO=INT(Y/8):LN=YAND7
120 BY=8192+RO*320+8*CH+LN:BI=7-(XAN
D7)
130 POKEBY,PEEK(BY)OR(2↑BI):NEXT:GOT
O110
140 INPUT"DAME[SPC]EL[SPC]SEMIEJE[SPC]
MAYOR";F
150 INPUT"DAME[SPC]EL[SPC]SEMIEJE[SPC]
MENOR";E
160 INPUT"DAME[SPC]LA[SPC]COORDENADA
[SPC]X[SPC]DEL[SPC]CENTRO";G
170 INPUT"DAME[SPC]LA[SPC]COORDENADA
[SPC]Y[SPC]DEL[SPC]CENTRO";H
180 SYS49152
190 FORS=0TO(2*π)STEP.03:X=G+(F*COS(
S)):Y=H+(E*SIN(S))
200 CH=INT(X/8):RO=INT(Y/8):LN=YAND7
:BY=8192+RO*320+8*CH+LN:BI=7-(XAND7)

210 POKEBY,PEEK(BY)OR(2↑BI):NEXTS:GO
TO190
220 PRINT"[CLR][COMM7]LA[SPC]EPICICL
OIDE[SPC]ES[SPC]CERRADA[SPC]SI[SPC]E
L[SPC]RADIO[SPC]DE[SPC]LA[SPC]CIRCUN
FERENCIA[SPC]FIJA"
230 PRINT"[CRSRU][24CRSRR]ES[SPC]MUL
TIPLO[SPC]DEL[SPC]DE[SPC]LA[SPC]CIRC
UNFERENCIA[SPC]MOVIL"
240 PRINT"ES[SPC]DECIR[SPC]RF=N*RM,E
N[SPC]ESTE[SPC]CASO[SPC]TENDRA[SPC]N
[2SPC]ARCOS"
250 PRINT"SI[SPC]SON[SPC]PRIMOS[SPC]
NO[SPC]CERRARA,[SPC]Y[SPC]SI[SPC]SON
[SPC]IGUA-LES[SPC]SERA[SPC]UNA[SPC]C
ARDIOIDE"
260 INPUT"DAME[SPC]EL[SPC]RADIO[SPC]
DE[SPC]LA[SPC]CIRCUNFERENCIA[SPC]FIJ
A";K
270 INPUT"DAME[SPC]EL[SPC]RADIO[SPC]
DE[SPC]LA[SPC]MOVIL,(MENOR[SPC]QUE[SPC]
EL[SPC]ANTERIOR)";L
280 SYS49152
290 FORT=0TO(2*π)STEP.03:X=150+((K+L
)*COS(T))-(L*COS(((K+L)/L)*T))
300 Y=100+((K+L)*SIN(T))-(L*SIN(((K+
L)/L)*T))
310 CH=INT(X/8):RO=INT(Y/8):LN=YAND7
:BY=8192+RO*320+8*CH+LN:BI=7-(XAND7)

320 POKEBY,PEEK(BY)OR(2↑BI):NEXTT
330 FORT=0TO(2*π)STEP.03:X=150+(K*CO
S(T)):Y=100+(K*SIN(T)):GOTO310
340 SYS49152
350 FORT=0TO(2*π)STEP.03:X=150+(50*C
OS(T)):Y=100+(50*SIN(T))
360 CH=INT(X/8):RO=INT(Y/8):LN=YAND7
:BY=8192+RO*320+8*CH+LN:BI=7-(XAND7)
```

```
370 POKEBY,PEEK(BY)OR(2↑BI):NEXT
380 FORT=(π/2)TO(3*π/2)STEP.03:X=150
-(25*COS(T)):Y=100-(50*SIN(T))
390 CH=INT(X/8):RO=INT(Y/8):LN=YAND7
:BY=8192+RO*320+8*CH+LN:BI=7-(XAND7)

400 POKEBY,PEEK(BY)OR(2↑BI):NEXT
410 FORT=(3*π/2)TO(π*2.5)STEP.1:X=15
0-(25*COS(T)):Y=100-(50*SIN(T))
420 CH=INT(X/8):RO=INT(Y/8):LN=YAND7
:BY=8192+RO*320+8*CH+LN:BI=7-(XAND7)

430 POKEBY,PEEK(BY)OR(2↑BI):NEXTT
440 FORT=0TOπSTEP.1:X=150-(50*COS(T)
):Y=100-(25*SIN(T))
450 CH=INT(X/8):RO=INT(Y/8):LN=YAND7
:BY=8192+RO*320+8*CH+LN:BI=7-(XAND7)

460 POKEBY,PEEK(BY)OR(2↑BI):NEXT
470 FORT=πTO(2*π)STEP.03:X=150-(50*C
OS(T)):Y=100-(25*SIN(T)):GOTO450
990 RESTORE:FORX=0TO77:READA:POKE491
52+X,A:NEXT
995 RETURN
1000 DATA173,24,208,9,8,141,24,208,1
73,17,208
1010 DATA9,32,141,17,208,169,32,133,
252,169,0
1020 DATA133,251,162,31,160,0,145,25
1,136,208,251
1030 DATA230,252,202,208,244,160,63,
145,251,136,16,251
1040 DATA169,0,133,253,169,4,133,254
,169,3,170
1050 DATA160,0,145,253,136,208,251,2
30,254,202,208
1060 DATA244,160,0,145,253,200,192,2
32,208,249,96
```

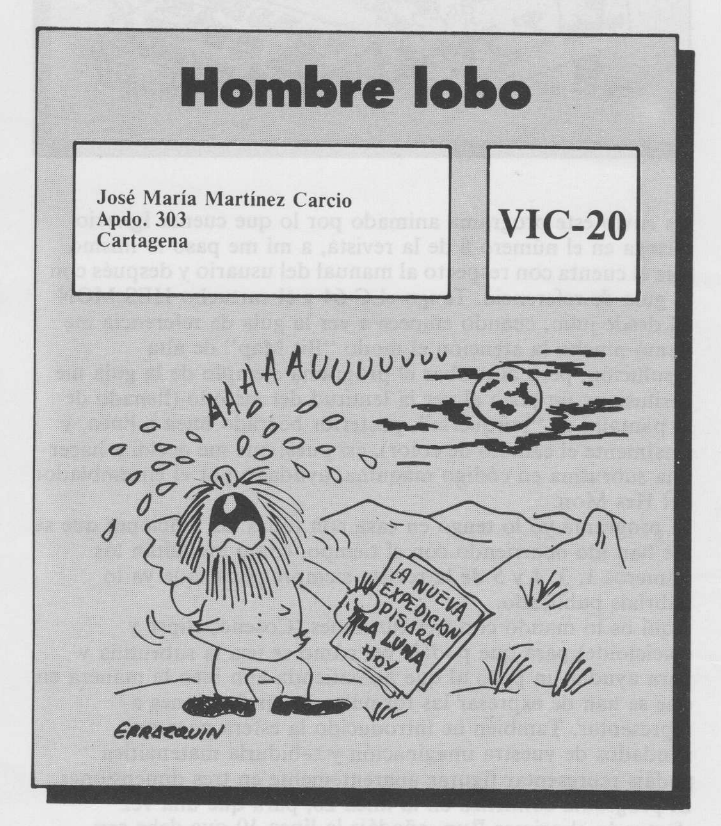
```
.,  C000   AD 18 D0 LDA $D018
.,  C003   09 08    ORA #$08
.,  C005   8D 18 D0 STA $D018
.,  C008   AD 11 D0 LDA $D011
.,  C00B   09 20    ORA #$20
.,  C00D   8D 11 D0 STA $D011
.,  C010   A9 20    LDA #$20
.,  C012   85 FC    STA $FC
.,  C014   A9 00    LDA #$00
.,  C016   85 FB    STA $FB
.,  C018   A2 1F    LDX #$1F
.,  C01A   A0 00    LDY #$00
.,  C01C   91 FB    STA ($FB),Y
.,  C01E   88       DEY
.,  C01F   D0 FB    BNE $C01C
.,  C021   E6 FC    INC $FC
.,  C023   CA       DEX
.,  C024   D0 F4    BNE $C01A
.,  C026   A0 3F    LDY #$3F
.,  C028   91 FB    STA ($FB),Y
```

```
.,  C02A   88       DEY
.,  C02B   10 FB    BPL $C028
.,  C02D   A9 00    LDA #$00
.,  C02F   85 FD    STA $FD
.,  C031   A9 04    LDA #$04
.,  C033   85 FE    STA $FE
.,  C035   A9 03    LDA #$03
.,  C037   AA       TAX
.,  C038   A0 00    LDY #$00
.,  C03A   91 FD    STA ($FD),Y
.,  C03C   88       DEY
.,  C03D   D0 FB    BNE $C03A
.,  C03F   E6 FE    INC $FE
.,  C041   CA       DEX
.,  C042   D0 F4    BNE $C038
.,  C044   A0 00    LDY #$00
.,  C046   91 FD    STA ($FD),Y
.,  C048   C8       INY
.,  C049   C0 E8    CPY #$E8
.,  C04B   D0 F9    BNE $C046
.,  C04D   60       RTS
.
```

Hombre lobo

José María Martínez Carcio
Apdo. 303
Cartagena

VIC-20



Parece ser que muchos vic-ciosos no se aclaran con el super-expander, otros no lo conocen y no saben las posibilidades de este con ahorro de memoria en unos comandos fáciles de aprender en vez de infinidad de pokes los cuales hay que saber. Así pues, me he animado a escribir este programa con una detallada explicación, para que los que no posean el super-expander o los que lean por primera vez la revista puedan enterarse.

10 fija el modelo de gráfica (alta resolución, a saber) y los colores que se emplea para la luna.

20-50 crean las variables a saber:

x.=coordenada horizontal del centro de las lunas

y.=coordenada vertical del centro

A=longitud del radio horizontal (0.7 factor de corrección para que la luna sea redonda)

b=longitud del radio vertical

100 imprime el círculo correspondiente

110-140 modifican las variables para que la luna suba y sea más pequeña
170-195 manera no muy ortodoxa de fijar las variables de la última luna
200 borra la pantalla
205-220 fija color, pinta la luna
235-280 cambia el color y pinta la figura
290-300 cambia el color y colorea por dentro la figura
310 música, a saber /ctrl/ /---/ = modelo música
v9=volumen
s3=voz
o3=octava de la voz
t3, t4, t5=tiempo
cdefgab=notas musicales (por orden)
320 borra la pantalla
330 para el programa

```
10 GRAPHIC2:COLOR0,6,1,1
20 X=512
30 Y=512
60 A=(500*0.7)
70 B=500
80 CIRCLE1,X,Y,A,B
90 X=X+50
100 Y=Y-50
110 A=A-(50*0.7)
120 B=B-50
150 IFY<=100THEN170
160 GOTO80
170 X=X-50
180 Y=Y+50
190 A=A+(50*0.7)
195 B=B+50
200 SCNCLR
205 COLOR6,0,1,0
210 CIRCLE1,X,Y,A,B
220 PAINT1,X,Y
240 COLOR0,6,1,1
250 DRAW1,100,870TO30,640TO65,430TO1
00,700TO380,280TO430,160TO400,54TO45
0,120
260 DRAW1,450,120TO512,74TO540,90TO6
15,30TO630,64TO580,128
270 DRAW1,580,128TO620,120TO530,240T
O478,364TO448,480TO448,832
280 DRAW1,448,832TO490,832TO490,870T
O410,870TO410,512TO280,670TO330,670
290 DRAW1,330,670TO330,832TO380,832T
O380,870TO100,870
300 COLOR6,0,0,1
310 PAINT1,45,640

310 PRINT"[CTRL ←]V9S3O3T3GGGCT4EGCT5E"
320 SCNCLR
330 END
```

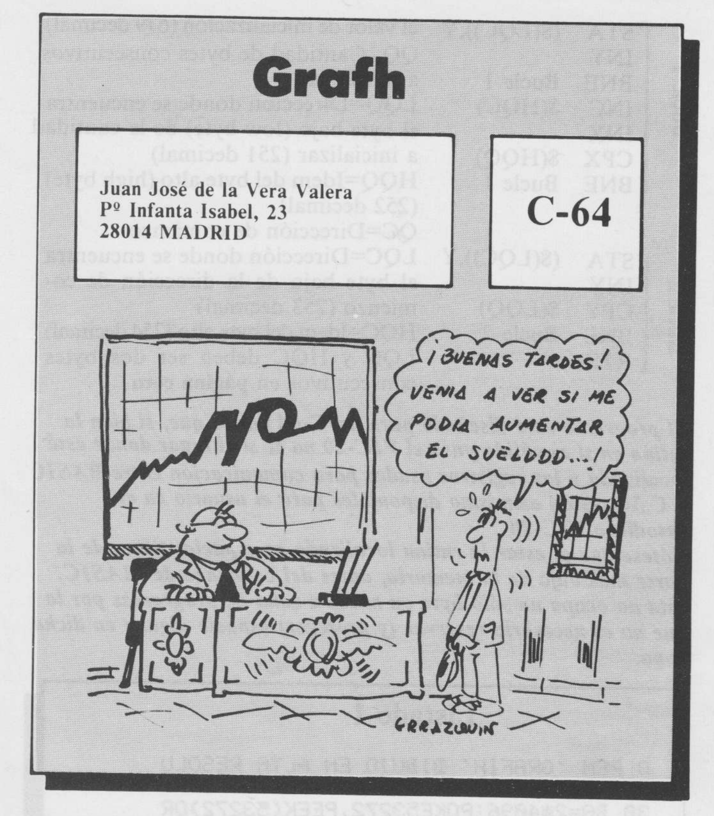


Directorio del cassette

Para ver el contenido del cassette:
LOAD"MXYZTPLK" (o un nombre similar)

Como este programa no lo tendrás en la cinta, el ordenador irá sacando en pantalla todos los nombres de los programas que encuentre.

D.L. RUN EE.UU.



Es éste mi primer contacto práctico en el C.M. Se trata de evitar la exasperante espera necesaria cuando pasamos a alta resolución y tenemos quę inicializar la memoria utilizada para el bitmapping (8000bytes). Con esta rutina esta labor se realiza en un parpadeo, con lo cual la desesperación se reduce a esperar que se pinte la figura a representar en la pantalla de alta resolución.

La rutina (27 bytes) la he localizado en un espacio libre en la página 2 (posiciones 680 a 706). Los registros de comunicación son:

679 .... valor de inicialización (0, en caso de limpieza de bit-map)
251, 252 .... cantidad de registros a inicializar
253, 254 .... dirección de comienzo

Veamos su aplicación con un ejemplo: se trata de pintar la famosa "huevera"; en la forma habitual el programa seria como el del Listado 1.

Incorporando el programa C.M. en forma de subrutina el programa quedaría como en el Listado 2.

Las líneas 1 y 2 contienen los datos del programa C.M. y la línea 10 lee el programa C.M. y lo introduce en 27 bytes consecutivos empezando en 680.

La subrutina Basic que comienza en 1000 calcula los valores a "pokear" en las posiciones 251 y 252 (cantidad de bytes a inicializar) y 253, 254 (dirección de comienzo). Se hacen los pokes, incluyendo el de la posición 679 con el valor de inicialización y después se llama a la subrutina C.M. (sentencia SYS680). Los registros de comunicación pueden cambiarse (modificando el programa C.M. convenientemente) pero es obligatorio que los que contienen la dirección de comienzo estén en la página cero.

Las variables QC, QQ, QI se utilizan para pasar los valores de dirección de comienzo (QC), cantidad de bytes consecutivos a inicializar (QQ) y valor de Inicialización (QI), a la subrutina 1000.

La nueva línea 51 pasa los valores 8192, 8000, 0 (limpieza de bit map) y la línea 71 pasa los valores 1024, 1000, 16 (inicialización de colores de fondo y trazo a negro y blanco).

Finalmente el listado en C.M. "sui generis" (o sea según mi entender) del programa.

```
LDA  $(DQI)      En donde:
LDY  #0          QI=Valor de Inicialización
LDX  #0          DQI=Dirección donde se encuentra
```

```
        ┌ STA   ($(LQC)),Y
        │ INY
        │ BNE   Bucle 1
Bucle 1 │ INC   $(HQC)
        │ INX
        │ CPX   $(HQQ)
        └ BNE   Bucle 1
        ┌ STA   ($(LQC)),Y
        │ INY
Bucle 2 │ CPY   $(LQQ)
        │ BNE   Bucle 2
        └ RTS
```

el valor de inicialización (679 decimal)
QQ=Cantidad de bytes consecutivos a inicializar.
LQQ=Dirección donde se encuentra el byte bajo (low byte) de la cantidad a inicializar (251 decimal)
HQQ=Idem del byte alto (high byte) (252 decimal)
QC=Dirección de comienzo
LQC=Dirección donde se encuentra el byte bajo de la dirección de comienzo (253 decimal)
HQC=Idem del byte alto (254 decimal)
LQC y HQC deben ser dos bytes consecutivos en **página cero**

El programa está diseñado para un C-64 por lo que, si bien la rutina en sí es válida para el VIC-20 no sé si el lugar donde está localizada y los registros usados para comunicación entre BASIC y C.M., están asimismo disponibles para el usuario en el susodicho VIC-20
Nótese que al estar la rutina localizada en espacios libres de la parte más baja de la memoria, antes del comienzo del BASIC, ésta no ocupa un sólo byte en nuestra zona de programas por lo que no es necesario reservar (y proteger) espacio alguno en dicha zona.

Listado 1

```
0 REM 'GRAF1H' DIBUJO EN ALTA RESOLU
CION (GRAFICO *HUEVERA*) JJV 09/84
30 BA=2*4096:POKE53272,PEEK(53272)OR
8:REM*SITUAR BIT MAP EN 8192
40 POKE53265,PEEK(53265)OR32:REM*ENT
RAR EN MODO BIT MAP
50 FORI=BATOBA+7999:POKEI,0:NEXT:REM
*BORRAR BIT MAP
60 POKE53280,0:REM*COLOR DE BORDE NE
GRO
70 FORI=1024TO2023:POKEI,16:NEXT:REM
*COLORES NEGRO(FONDO) Y BLANCO(TRAZO
)
80 X0=160:Z0=200:REM*CENTRADO DEL GR
AFICO
90 R=10:FORX1=-100TO100:J=0
100 V=R*INT(SQR(10000-X1*X1)/R)
110 FORY=VTO-VSTEP-R
120 Z1=INT(80+30*SIN(SQR(X1*X1+Y*Y)/
12)-.7*Y)
130 IFZ1<JTHEN180
135 J=Z1:X=X1+X0:Z=Z0-Z1
140 RO=INT(Z/8):CH=INT(X/8):LI=ZAND7

150 BI=7-(XAND7)
160 BY=BA+RO*320+CH*8+LI
170 POKEBY,PEEK(BY)OR2↑BI
180 NEXTY:NEXTX1
190 GETA$:IFA$=""THEN190
200 POKE53265,PEEK(53265)AND223:REM*
MODO BIT MAP OFF
210 POKE53272,PEEK(53272)AND247
220 PRINT"[CLR]":END
```

Listado 2

```
0 REM 'GRAF2H' DIBUJO EN ALTA RESOLU
CION (GRAFICO *HUEVERA*) JJV 09/84
1 DATA173,167,2,160,0,162,0,145,253,
200,208,251,230,254,232,228,252
2 DATA208,244,145,253,200,196,251,20
8,249,96
10 FORI=0TO26:READDD:POKE680+I,DD:NE
XT:REM* LECTURA DE PROGRAMA C.M.
30 BA=2*4096:POKE53272,PEEK(53272)OR
8:REM*SITUAR BIT MAP EN 8192
40 POKE53265,PEEK(53265)OR32:REM*ENT
RAR EN MODO BIT MAP
51 QC=BA:QQ=8000:QI=0:GOSUB1000:REM*
 BORRAR BIT MAP
60 POKE53280,0:REM*COLOR DE BORDE NE
GRO
71 QC=1024:QQ=1000:QI=16:GOSUB1000:R
EM* INICIALIZAR COLORES DE FONDO-TRA
ZO
80 X0=160:Z0=200:REM*CENTRADO DEL GR
AFICO
90 R=10:FORX1=-100TO100:J=0
100 V=R*INT(SQR(10000-X1*X1)/R)
110 FORY=VTO-VSTEP-R
120 Z1=INT(80+30*SIN(SQR(X1*X1+Y*Y)/
12)-.7*Y)
130 IFZ1<JTHEN180
135 J=Z1:X=X1+X0:Z=Z0-Z1
140 RO=INT(Z/8):CH=INT(X/8):LI=ZAND7

150 BI=7-(XAND7)
160 BY=BA+RO*320+CH*8+LI
170 POKEBY,PEEK(BY)OR2↑BI
180 NEXTY:NEXTX1
190 GETA$:IFA$=""THEN190
200 POKE53265,PEEK(53265)AND223:REM*
MODO BIT MAP OFF
210 POKE53272,PEEK(53272)AND247
220 PRINT"[CLR]":END
1000 REM* SUBRUTINA DE INICIALIZACIO
N DE UNA ZONA DE MEMORIA
1010 POKE251,QQ-INT(QQ/256)*256:POKE
252,INT(QQ/256):REM* NUM. DE BYTES A
 INIC.
1020 POKE253,QC-INT(QC/256)*256:POKE
254,INT(QC/256):REM* DIRECCION DE CO
MIENZO
1030 POKE679,QI:REM* VALOR DE INICIA
LIZACION
1040 SYS680:RETURN
```



Generador de sonidos

He aquí un pequeño programa en lenguaje máquina que hace sonar las teclas cada vez que se pulsan:

```
60000 fora=828to861:readb:pokea,b:next
60010 data 169,15,141,14,144,120,169,78,141,20
60020 data 3,169,3,141,21,3,88,96,165,197
60030 data 201,128,240,7,101,197,105,128,141,12
60040 data 144,76,191,234
```

Para hacerlo funcionar hay que hacer SYS 828, y para pararlo, pulsar STOP.

T.J.C. RUN EE.UU.

Caracteres indocumentados

Además de los caracteres que aparecen en el teclado, hay otros cuatro que no se ven. Son los siguientes: ASCIIs 126,127,169,186 (en pokes 94,95,105,122).

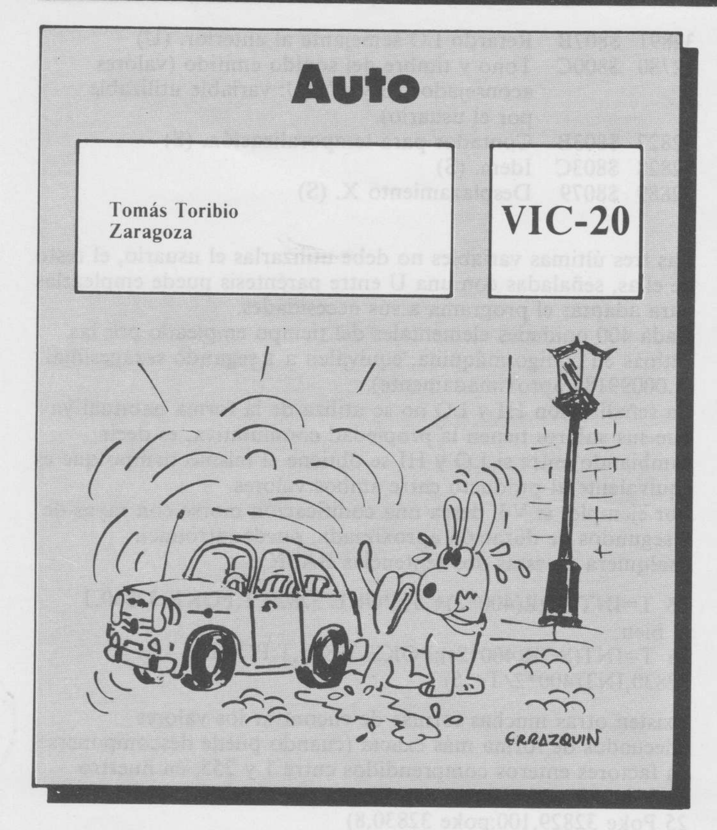
Corresponden a pulsar SHIFT P, COMM *, SHIFT £, SHIFT @, estando en modo minúsculas.

S.C. RUN EE.UU.

En una linea

```
1 n=(rnd(1)*16)+1:b$=mid$("—texto—",n,1):print spc(n)b$
"[CRSR U]":goto1
```

P.S. RUN EE.UU.



Auto

Tomás Toribio
Zaragoza

VIC-20

Se trata de una respuesta a Peré Esquerra, que en la sección de Carta Blanca del número 5 (junio) preguntaba si se podía usar la función AUTO sin tener el cartucho de "Ayuda al programador".

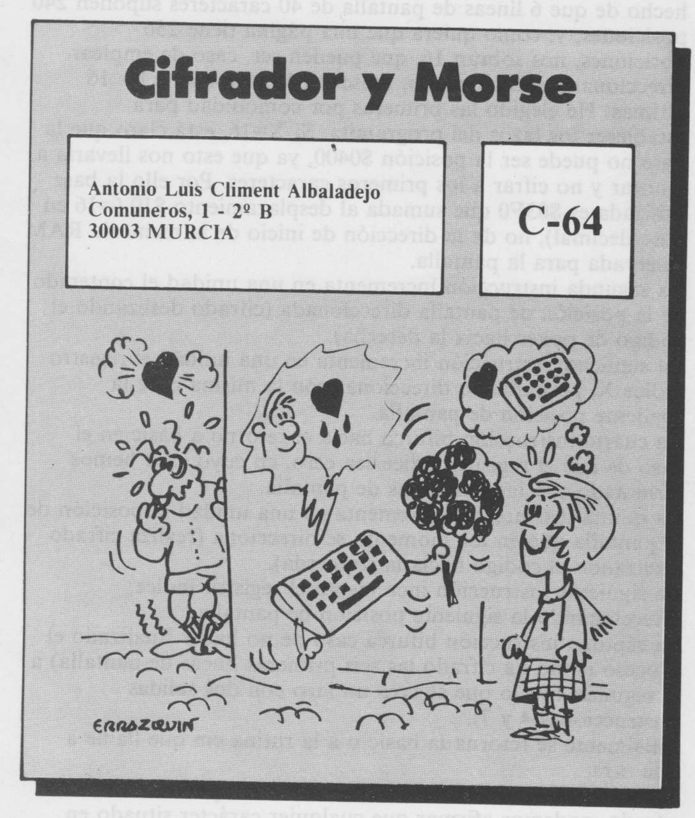
Pues este programa-rutina equivale a la función AUTO. Al teclearlo hay que respetar la numeración de las líneas. Una vez escrito hay que grabarlo, porque al final tiene una rutina de autodestrucción.

Al hacer RUN pregunta el número de la primera línea del programa. A continuación el incremento o diferencia entre las siguientes líneas. La entrada de datos se hace mediante un GET (línea 63992), el cursor parpadea gracias al POKE 204,0 de la línea anterior. El parpadeo es a veces bastante irregular, debido al POKE. Para salir de la rutina entrar ← y RETURN. Las líneas 63990 a 63997 desaparecerán quedando solamente el programa creado.

Espero que el programa os parezca interesante y que podáis publicarlo.

Saludos de un vic-cioso

```
0 REM *********************************
***
1 REM ** FUNCION AUTO PARA EL VIC-20
 **
2 REM ** TOMAS TORIBIO. ZARAGOZA
 **
3 REM *********************************
***
63990 INPUT"[CLR]LINEA[SPC]INICIAL";
A:INPUT"INCREMENTO";B:POKE2,B:PRINT"
[CLR]";
63991 B=A/256:POKE0,(B-INT(B))*256:P
OKE1,B:PRINTMID$(STR$(A),2);:POKE204
,0
63992 GETA$:IFA$="←"THENPRINT"[CLR]"
;:FORI=0TO7:PRINT63990+I:POKE632+I,1
3:NEXT:GOTO63997
63993 PRINTA$;:IFA$<>CHR$(13)THEN639
92
63994 PRINT"G[SHIFTO]63995":FORA=631
TO634:POKEA,145:NEXT:POKEA,13:POKEA+
1,13:POKE198,6:END
63995 PRINT"[2CRSRU]":FORA=1TO3:PRIN
T"[7SPC]":NEXT:PRINT"[3CRSRU]";
63996 A=PEEK(0)+256*PEEK(1)+PEEK(2):
GOTO63991
63997 POKE631,19:POKE640,13:POKE198,
10:PRINT"L[SHIFTI]":END
```



Cifrador y Morse

Antonio Luis Climent Albaladejo
Comuneros, - 2º B
30003 MURCIA

C-64

Cifrador 1084

Sirve para cifrar o descifrar el mensaje contenido en las seis primeras líneas de pantalla. Las líneas inferiores de la pantalla se emplean para dar al usuario las instrucciones oportunas sobre el manejo del programa.

Morse 1084

Emite el sonido equivalente al mensaje situado en las primeras seis líneas de pantalla, codificado en MORSE. El resto de las líneas de pantalla tienen la misma función que en el programa anterior. Es posible cambiar el valor de determinadas posiciones (como ya se explicará más adelante), logrando distinto tono y timbre y fijar los tiempos de duración del punto y raya y separación entre caracteres. Jugando con estos tiempos pueden lograrse dispositivos que activen de forma automática y a gran distancia sistemas de alarmas (como ocurre con los radiofaros).

Morse-C 1084

Realiza las funciones de los dos anteriores, con lo que resulta posible emitir en morse un mensaje cifrado. El cifrador utilizado en este programa es distinto a la primera rutina ya que debe restringir el juego de caracteres válidos únicamente a los alfabéticos. Por otra parte se simplifica al limitarse sin deslizar el alfabeto en uno u otro sentido tantas veces como se pulse la tecla de función adecuada.

Descripción de las rutinas escritas en ensamblador utilizadas por los programas.

Cifrador 1084. Consta de 16 bytes y siete instrucciones. La dirección de inicio de pantalla es la $0400, por lo que podemos

direccionar a ella como si se tratara de una tabla, utilizando cualquiera de los registros índice del procesador. La posición $8000, la inmediata anterior al comienzo de la rutina contendrá el único valor utilizado como variable por la rutina cifradora, por el motivo que descubrirá el propio lector al hablar del descifrado.
La primera instrucción carga en el registro índice X, el valor contenido en la posición $8000 y que deberá ser dieciséis para direccionar al inicio de la pantalla. Este valor inicial se debe al hecho de que 6 líneas de pantalla de 40 caracteres suponen 240 posiciones, y, como quiera que una página tiene 256 posiciones, nos sobran 16, que pueden ser, caso de emplear direccionamiento indexado, tanto las 16 primeras o las 16 últimas. He elegido las primeras por comodidad para establecer los lazos del programita. Si X=16, está claro que la base no puede ser la posición $0400, ya que esto nos llevaría a ignorar y no cifrar a los primeros caracteres. Por ello la base utilizada es $03F0 que sumada al desplazamiento $10 (=16 en base decimal), no da la dirección de inicio de la memoria RAM reservada para la pantalla.
La segunda instrucción incrementa en una unidad el contenido de la posición de pantalla direccionada (cifrado deslizando el código de pokes hacia la derecha).
La siguiente instrucción incrementa en una unidad el registro índice X, permitiendo direccionar con la misma base la siguiente posición de pantalla.
La cuarta instrucción bifurca hacia el retorno a basic en el caso de que el registro índice sea cero, en cuyo caso hemos terminado con las seis líneas de pantalla.
La quinta instrucción decrementa en una unidad la posición de la pantalla que en ese momento se direcciona (realiza cifrado deslizando el código hacia la izquierda).
La siguiente instrucción incrementa el registro índice, direccionando la siguiente posición de pantalla.
La séptima instrucción bifurca caso de no haber finalizado el proceso (si no ha cifrado las seis primeras líneas de pantalla) a la segunda con lo que se crea un lazo con dos salidas (instrucciones 4 y 7).
Finalmente se retorna la basic o a la rutina cm que llame a esta otra.
Como conclusión, ante el examen sobre la operación de cifrado, podemos afirmar que cualquier carácter situado en una posición impar de la pantalla (dirección de memoria RAM par) sufre un cifrado pasando a ser el siguiente carácter de la lista poke que figura en el manual del usuario; y el que ocupe una posición par de pantalla e impar de memoria pasa a ser el carácter anterior de la lista de códigos poke mencionada. En otras palabras una C sufre una transición hacia D o B dependiendo de su posición impar o par de pantalla. Para el cifrado, la salida de la cuarta instrucción es inútil, ya que al incrementar en uno una dirección par, jamás puede conseguirse un par, y cero es par. Sin embargo es útil si se emplea la misma rutina para descifrar. ¿Cómo? Si el desplazamiento inicial no es $10 (16) sino $0F (15) el inicio de la exploración no será la primera posición de pantalla, sino la anterior (sin uso según el manual del usuario) la cual incrementara, la 2ª, si que coincide con la primera de pantalla, pero en esta ocasión la decrementará realizando, por tanto, el proceso contrario al anterior, es decir, el descifrado.

Morse 1084. Una descripción tan detallada como la efectuada para el programa anterior superaría los límites de espacio que, me imagino, puede conceder la revista. Por ello será bastante más superficial, entrando más a fondo en aquellos detalles que merezcan la pena.
Las distintas rutinas en código máquina utilizadas y el conjunto de variables y constantes por ellas empleadas ocupan 205 bytes, distribuidos de la siguiente forma:

32768	$8000	Duración HI del punto. (U)
32769	$8001	Duración LO del punto. (U)
32829	$803D	Duración HI de la raya. (U)
32830	$803E	Duración LO de la raya. (U)
32890	$807A	Retardo HI para caracteres no alfabéticos, incluido el espacio. (U)
32891	$807B	Retardo LO semejante al anterior. (U)
32780	$800C	Tono y timbre del sonido emitido (valores aconsejados 33 y 17) (U: variable utilizable por el usuario).
32827	$803B	Contador para temporalización. (S)
32828	$803C	Idem. (S)
32889	$8079	Desplazamiento X. (S)

Las tres últimas variables no debe utilizarlas el usuario, el resto de ellas, señaladas con una U entre paréntesis puede emplearlas para adaptar el programa a sus necesidades.
Cada 400 unidades elementales del tiempo empleado por las rutinas en código máquina, equivalen a 1 segundo sexagesimal (1.00099184 aproximadamente).
La señalización HI y LO no se utiliza de la forma habitual ya que sus valores tienen la propiedad conmutativa, es decir, cambiando entre si LO y HI se obtiene el mismo tiempo que es equivalente al producto entre ambos valores.
Por ejemplo, si Vd. desea una codificación morse con rayas de 2 segundos de duración aproximada, puede introducir cualquiera de estas dos sentencias BASIC.

```
25 T=INT(SQR(400*2)+.5):POKE 32829,T:POKE 32830,T
O bien,
25 T=INT(SQR(400*2)):POKE 32829,T:POKE
32830,INT(400*2/T+.5)
```

Existen otras muchas formas de encontrar los valores adecuados de forma más exacta (cuando puede descomponerse en factores enteros comprendidos entre 1 y 255, en nuestro caso 400*2=800=100*8, con lo que puede escribirse:
25 Poke 32829,100:poke 32830,8)

CONSTANTES

32863	$805F	Inicio tabla código morse (corresponde a la letra A)
....	....	... ...
32888	$8078	Fin tabla código morse (corresponde a la letra Z)

RUTINAS terminadas con la instrucción RTS

32770	$8002	Prepara SID con el sonido elegido ($800C), volumen 0
32795	$801B	Emite sonido de duración ($8000)*($8001), correspondiente al punto morse (volumen 15)
32802	$8022	Realiza pausa (silencio) equivalente a la duración del punto.
32831	$803F	Emite sonido de duración ($803D)*($803E) correspondiente al signo raya del alfabeto Morse.
32838	$8046	Realiza pausa (silencio) equivalente a la duración de una raya.
32892	$807C	Efectúa codificación Morse, emitiendo los sonidos equivalentes al mensaje situado en las 6 primeras líneas de pantalla.

Esta última rutina emplea a todas las demás, excepto la rutina de preparación del SID, que se utiliza directamente desde el programa codificado en BASIC. Su funcionamiento en grandes líneas es el siguiente: Explora el área de memoria RAM correspondiente a las 6 primeras líneas, si el carácter encontrado no es válido (no es alfabético) realiza una pausa de silencio equivalente a ($807A)*($807B), caso contrario (el carácter es alfabético) toma de la tabla de códigos Morse el que le corresponda llamando a las rutinas de emisión de punto o raya según código y dejando entre signo y signo morse (pe. entre un punto y una raya) un silencio equivalente al punto. Cuando se detectan dos caracteres válidos consecutivos (que forman parte de la misma palabra) se realiza un silencio equivalente a la duración de una raya.

La tabla de códigos morse contenida a partir de las posiciones $805F es la siguiente:

Letra Alfabeto	Código Morse	Representación binaria				Valor decimal
A	.—	01	10	00	00	96
B	—...	10	01	01	01	149
C	—.—.	10	01	10	01	153
D	—..	10	01	01	00	148
E	.	01	00	00	00	64
F	..—.	01	01	10	01	89
G	— —.	10	10	01	00	164
H	....	01	01	01	01	85
I	..	01	01	00	00	80
J	.— — —	01	10	10	10	106
K	—.—	10	01	10	00	152
L	.—..	01	10	01	01	101
M	——	10	10	00	00	160
N	—.	10	01	00	00	144
O	— — —	10	10	10	00	168
P	.— —.	01	10	10	01	105
Q	— —.—	10	10	01	10	166
R	.—.	01	10	01	00	100
S	...	01	01	01	00	84
T	—	10	00	00	00	128
U	..—	01	01	10	00	88
V	...—	01	01	01	10	86
W	.— —	01	10	10	00	104
X	—..—	10	01	01	10	150
Y	—.— —	10	01	10	10	154
Z	— —..	10	10	01	01	165

Como el lector habrá observado para cada letra se emplea un byte y para cada signo del alfabeto morse dos bit, con lo que en cada byte pueden existir hasta 4 signos Morse que es la longitud máxima de una letra codificada en Morse.
Los pares de bit, tienen el siguiente significado:

10 Raya
01 Punto
00 Fin carácter Morse.

Cuando la rutina codificadora toma un código Morse binario de la tabla correspondiente examina su contenido rotándolo hacia la izquierda y llama a las rutinas de emisión de sonido que correspondan.
Puede adaptarse un conjunto de relés (son suficientes dos salidas) para conectar directamente el ordenador a una emisora decamétrica para emisión en código Morse.

"MORSE-C 1084". Es el programa más complejo.
Todo lo que se ha dicho respecto a "MORSE 1084" es aplicable a esta versión. Las rutinas máquinas son idénticas, pero se han añadido dos más que suponen unos 61 bytes adicionales. Estas rutinas son:

32973	$80CD	Rota abecedario 1 posición hacia la izquierda B efectuará transición hacia A y esta hacia Z.
33002	$80EA	Rota abecedario 1 posición hacia la izquierda.

La rutina es algo más complicada a la del primer programa debido a que sólo se cifran los caracteres alfabéticos y a que es preciso establecer una continuidad entre la z y la a, y viceversa, esto es detrás de la z se inicia de nuevo el abecedario.
Se adjuntan los listados en ensamblador a fin de que sea posible estudiar las rutinas descritas.

Cifrador 1084

```
10 D=32769:FORI=D TO D+15:READ J:POK
E I,J:NEXT
30 PRINT"[CLR][7CRSRD][BLK][RVSON]ES
TE[SPC]PROGRAMA[SPC]SIRVE[SPC]PARA[SPC]
CIFRAR[SPC]O[SPC]DESCI-";
40 PRINT"[RVSON]FRAR[SPC]LOS[SPC]CAR
ACTERES[SPC]ESCRITOS[SPC]EN[SPC]LAS[SPC]
SEIS";
```

```
50 PRINT"[RVSON]PRIMERAS[SPC]LINEAS[SPC]
DE[SPC]PANTALLA.[SPC]COMANDOS:[GRN][5CRSRD]
"
60 PRINT"HOME/CLR[SPC]:[2SPC]POSICIO
NARSE[SPC]INICIO/BORRAR"
70 PRINT"INST/DEL[SPC]:[2SPC]INSERTA
R[SPC]/[SPC]DELETEAR[SPC]CARACTER";
80 PRINT"CURSORES[SPC]:[2SPC]MOVIMIE
NTO[SPC]CURSOR"
90 PRINT"F1[7SPC]:[2SPC]CIFRAR[SPC]"

100 PRINT"F3[7SPC]:[2SPC]DESCIFRAR[SPC]
"
110 PRINT"F5[7SPC]:[2SPC]IMPRIMIR[SPC]
"
120 PRINT"F7[7SPC]:[2SPC]FIN[2SPC]"
130 PRINT"TECLADO[2SPC]:[2SPC]USO[SPC]
COMUN[SPC]"
140 PRINT"[HOM]";
150 GOSUB 240
160 GETA$:IFA$=""THEN150
170 IF ASC(A$)=133THEN POKE 32768,16
:SYS D:GOTO 160
180 IF ASC(A$)=134THEN POKE 32768,15
:SYS D:GOTO 160
190 IF ASC(A$)=135THEN 270
200 IF ASC(A$)=136THEN END
210 IF ASC(A$)=147 THEN 30
220 PRINTA$;:GOTO 160
230 STOP
240 T=80:PRINT"[RVSON][BLK][SPC][CRSRL]
";:FOR I=1 TO T:NEXT
250 PRINT"[RVSOFF][COMM7][SPC][CRSRL]
";:FOR I=1 TO T:NEXT
260 RETURN
270 L$="":FOR C=1024 TO1264:P=PEEK(C
)
280 IFP>190THEN B$="[SPC]":GOTO 310
290 B$=CHR$(((P+64)AND(P<32))+((P+32
)AND(P>63))+(P AND (P>32) AND (P<64)
))
300 IF ASC(B$)=0THEN B$="[SPC]"
310 L$=L$+B$:NEXT C
320 OPEN4,4:PRINT#4,L$:CLOSE4:GOTO 1
60
330 STOP
10000 DATA 174, 0, 128, 254, 240, 3,
 232, 240
10010 DATA 6, 222, 240, 3, 232, 208,
 244, 96
```

```
RUTINA ENSAMBLADOR CIFRADOR 1084

.

.,   8001 AE 00 80 LDX $8000
.,   8004 FE F0 03 INC $03F0,X
.,   8007 E8       INX
.,   8008 F0 06    BEQ $8010
.,   800A DE F0 03 DEC $03F0,X
.,   800D E8       INX
.,   800E D0 F4    BNE $8004
.,   8010 60       RTS
```

Morse 1084

```
10 S=32768:FOR I=S TOS+204:READ A:PO
KEI,A:SU=SU+A:NEXT
20 IF SU<> 25071 THEN PRINT"[CLR][3CRSRD]
ERROR[SPC]SENTENCIAS[SPC]DATA"
30 SYS 32770:T=80
40 PRINT"[CLR][6CRSRD][CYN][40COMMU]
";
50 PRINT"[BLK]ESTE[SPC]PROGRAMA[SPC]
GENERA[SPC]SONIDO[SPC]CORRESPON--";
```

```
60 PRINT"DIENTE[SPC]AL[SPC]CODIGO[SPC]
MORSE[SPC]DE[SPC]UN[SPC]MENSAJE[SPC]
DE[SPC]";
70 PRINT"240[SPC]CARACTERES[SPC]COMO
[SPC]MAXIMO[SPC]DE[SPC]LONGITUD[SPC]
/";
80 PRINT"SITUADO[SPC]EN[SPC]LAS[SPC]
SEIS[SPC]PRIMERAS[SPC]LINEAS[SPC]DE[SPC]
/";
90 PRINT"PANTALLA."
100 PRINT"SOLO[SPC]SON[SPC]VALIDOS[SPC]
LOS[SPC]CARACTERES[SPC]ALFABE--";
110 PRINT"TICOS[SPC]IGNORANDOSE[SPC]
EL[SPC]RESTO."
120 PRINT"[CRSRD][RVSON][6SPC]PULSE[SPC]
F1[SPC]PARA[SPC]CODIFICAR[SPC]MORSE[5SPC]
";
130 PRINT"[RVSON][6SPC]PULSE[SPC]FX[SPC]
,[SPC]X<>1[SPC]PARA[SPC]TERMINAR[5SPC]
[RVSOFF]";
140 PRINT"[CRSRD]ESTUDIANDO[SPC]LAS[SPC]
INSTRUCCIONES[SPC]DEL[SPC]PROGRA-";
150 PRINT"MA[SPC]PODRA[SPC]REGULAR[SPC]
LA[SPC]DURACION[SPC]DEL[SPC]PUNTO,[SPC]
";
160 PRINT"COMA[SPC]Y[SPC]PAUSA[SPC]P
OR[SPC]BLANCO[SPC]O[SPC]CARACTER[SPC]
DES--";
170 PRINT"CONOCIDO."
180 PRINT"[CYN][CRSRD][40COMMO]";
190 PRINT"[HOM]";
200 PRINT"[BLK][RVSON][SPC][CRSRL]";
:FORI=1TOT:NEXT:PRINT"[COMM7][RVSOFF]
[SPC][CRSRL]";:FOR I=1TO T:NEXT
210 GET A$:IF A$=""THEN 200
220 IF ASC(A$)=133 THEN SYS 32892:GO
TO 200
230 IF ASC(A$)>133 AND ASC(A$)<141TH
EN PRINT:PRINT"FIN[SPC]PROGRAMA":END
240 IF ASC(A$)=19 THEN 190
250 IF ASC(A$)=147 THEN 40
260 PRINT A$;:GOTO 200
270 DATA 2, 20, 169, 255, 162, 6, 15
7, 0
280 DATA 212, 202, 202, 169, 33, 157
, 0, 212
290 DATA 202, 202, 202, 157, 0, 212,
 202, 157
300 DATA 0, 212, 96, 162, 24, 169, 1
5, 157
310 DATA 0, 212, 172, 0, 128, 174, 1
, 128
320 DATA 206, 59, 128, 208, 251, 202
, 208, 248
330 DATA 136, 208, 242, 162, 24, 169
, 0, 157
340 DATA 0, 212, 96, 0, 0, 10, 20, 1
62
350 DATA 24, 169, 15, 157, 0, 212, 1
72, 61
360 DATA 128, 174, 62, 128, 206, 60,
 128, 208
370 DATA 251, 202, 208, 248, 136, 20
8, 242, 162
380 DATA 24, 169, 0, 157, 0, 212, 96
, 96
390 DATA 149, 153, 148, 64, 89, 164,
 85, 80
400 DATA 106, 152, 101, 160, 144, 16
8, 105, 166
410 DATA 100, 84, 128, 88, 86, 104,
150, 154
420 DATA 165, 0, 169, 16, 169, 16, 1
41, 121
430 DATA 128, 174, 121, 128, 189, 24
0, 3, 168
440 DATA 136, 24, 105, 0, 48, 45, 24
, 233
450 DATA 26, 16, 40, 185, 95, 128, 7
2, 41
460 DATA 192, 240, 28, 104, 72, 41,
128, 208
470 DATA 16, 32, 27, 128, 32, 34, 12
8, 104
480 DATA 24, 42, 42, 41, 252, 72, 18
4, 80
490 DATA 230, 32, 63, 128, 184, 80,
237, 32
500 DATA 70, 128, 104, 172, 122, 128
, 174, 123
510 DATA 128, 202, 208, 253, 136, 20
8, 247, 238
520 DATA 121, 128, 208, 181, 96
```

```
.,  8002 A9 FF     LDA #$FF
.,  8004 A2 06     LDX #$06
.,  8006 9D 00 D4  STA $D400,X
.,  8009 CA        DEX
.,  800A CA        DEX
.,  800B A9 21     LDA #$21
.,  800D 9D 00 D4  STA $D400,X
.,  8010 CA        DEX
.,  8011 CA        DEX
.,  8012 CA        DEX
.,  8013 9D 00 D4  STA $D400,X
.,  8016 CA        DEX
.,  8017 9D 00 D4  STA $D400,X
.,  801A 60        RTS
.,  801B A2 18     LDX #$18
.,  801D A9 0F     LDA #$0F
.,  801F 9D 00 D4  STA $D400,X
.,  8022 AC 00 80  LDY $8000
.,  8025 AE 01 80  LDX $8001
.,  8028 CE 3B 80  DEC $803B
.,  802B D0 FB     BNE $8028
.,  802D CA        DEX
.,  802E D0 F8     BNE $8028
.,  8030 88        DEY
.,  8031 D0 F2     BNE $8025
.,  8033 A2 18     LDX #$18
.,  8035 A9 00     LDA #$00
.,  8037 9D 00 D4  STA $D400,X
.,  803A 60        RTS
.,  803B 00        BRK
.,  803C 00        BRK
.,  803D 02        ???
.,  803E 14        ???
.,  803F A2 18     LDX #$18
.,  8041 A9 0F     LDA #$0F
.,  8043 9D 00 D4  STA $D400,X
.,  8046 AC 3D 80  LDY $803D
.,  8049 AE 3E 80  LDX $803E
.,  804C CE 3C 80  DEC $803C
.,  804F D0 FB     BNE $804C
.,  8051 CA        DEX
.,  8052 D0 F8     BNE $804C
.,  8054 88        DEY
.,  8055 D0 F2     BNE $8049
.,  8057 A2 18     LDX #$18
.,  8059 A9 00     LDA #$00
.,  805B 9D 00 D4  STA $D400,X
.,  805E 60        RTS
```

Morse-C 1084

```
10 S=32768:FOR I=S TOS+267:READ A:PO
KEI,A:SU=SU+A:NEXT
20 IF SU<> 32871 THEN PRINT"[CLR][3CRSRD]
ERROR[SPC]SENTENCIAS[SPC]DATA"
30 SYS 32770:T=80
40 PRINT"[CLR][6CRSRD][CYN][40COMMU]
";
```

```
50 PRINT"[BLK]ESTE[SPC]PROGRAMA[SPC]
GENERA[SPC]SONIDO[SPC]CORRESPON---";

60 PRINT"DIENTE[SPC]AL[SPC]CODIGO[SPC]
MORSE[SPC]DE[SPC]UN[SPC]MENSAJE[SPC]
DE[SPC]";
70 PRINT"240[SPC]CARACTERES[SPC]COMO
[SPC]MAXIMO[SPC]DE[SPC]LONGITUD[SPC]
/";
80 PRINT"SITUADO[SPC]EN[SPC]LAS[SPC]
SEIS[SPC]PRIMERAS[SPC]LINEAS[SPC]DE[SPC]
/";
90 PRINT"PANTALLA."
100 PRINT"SOLO[SPC]SON[SPC]VALIDOS[SPC]
LOS[SPC]CARACTERES[SPC]ALFABE--";
110 PRINT"TICOS[SPC]IGNORANDOSE[SPC]
EL[SPC]RESTO."
120 PRINT"[CRSRD]EL[SPC]CIFRADO,[SPC]
DESCIFRADO[SPC]SE[SPC]PRODUCE[SPC]DE
SLI-";
130 PRINT"ZANDO[SPC]EL[SPC]ALFABETO[SPC]
A[SPC]IZQUIERDA[SPC]O[SPC]DERECHA.";

140 PRINT"[COMM1][CRSRD][RVSON]F1[SPC]
:[SPC]PARA[SPC]CODIFICAR[SPC]MORSE[15SPC]
";
150 PRINT"[RVSON]F3[SPC]:[SPC]CIFRAR
,[SPC]DESPLAZA[SPC]IZQUIERDA[SPC]MEN
SAJE[SPC]";
160 PRINT"[RVSON]F5[SPC]:[SPC]DESCIF
RAR,[SPC]DESPLAZA[SPC]DERECHA[SPC]ME
NSAJE";
170 PRINT"[RVSON]F7[SPC]:[SPC]TERMIN
AR[27SPC]";
180 PRINT"[CYN][CRSRD][40COMMO]";
190 PRINT "[HOM]";
200 PRINT"[BLK][RVSON][SPC][CRSRL]";
:FORI=1TOT:NEXT:PRINT"[COMM7][RVSOFF]
[SPC][CRSRL]";:FOR I=1TO T:NEXT
210 GET A$:IF A$=""THEN 200
220 IF ASC(A$)=133 THEN SYS 32892:GO
TO 200
230 IF ASC(A$)=134 THEN SYS 32973:GO
TO 200
240 IF ASC(A$)=135 THEN SYS 33002:GO
TO 200
250 IF ASC(A$)>135 AND ASC(A$)<141TH
EN PRINT:PRINT"FIN[SPC]PROGRAMA":END

260 IF ASC(A$)=19 THEN 190
270 IF ASC(A$)=147 THEN 40
280 PRINT A$;:GOTO 200
290 DATA 2, 20, 169, 255, 162, 6, 15
7, 0
300 DATA 212, 202, 202, 169, 33, 157
, 0, 212
310 DATA 202, 202, 202, 157, 0, 212,
 202, 157
320 DATA 0, 212, 96, 162, 24, 169, 1
5, 157
330 DATA 0, 212, 172, 0, 128, 174, 1
, 128
340 DATA 206, 59, 128, 208, 251, 202
, 208, 248
350 DATA 136, 208, 242, 162, 24, 169
, 0, 157
360 DATA 0, 212, 96, 0, 0, 10, 20, 1
62
370 DATA 24, 169, 15, 157, 0, 212, 1
72, 61
380 DATA 128, 174, 62, 128, 206, 60,
 128, 208
390 DATA 251, 202, 208, 248, 136, 20
8, 242, 162
400 DATA 24, 169, 0, 157, 0, 212, 96
, 96
410 DATA 149, 153, 148, 64, 89, 164,
 85, 80
420 DATA 106, 152, 101, 160, 144, 16
8, 105, 166
430 DATA 100, 84, 128, 88, 86, 104,
150, 154
440 DATA 165, 0, 169, 16, 169, 16, 1
41, 121
450 DATA 128, 174, 121, 128, 189, 24
0, 3, 168
460 DATA 136, 24, 105, 0, 48, 45, 24
, 233
470 DATA 26, 16, 40, 185, 95, 128, 7
2, 41
480 DATA 192, 240, 28, 104, 72, 41,
128, 208
490 DATA 16, 32, 27, 128, 32, 34, 12
8, 104
500 DATA 24, 42, 42, 41, 252, 72, 18
4, 80
510 DATA 230, 32, 63, 128, 184, 80,
237, 32
520 DATA 70, 128, 104, 172, 122, 128
, 174, 123
530 DATA 128, 202, 208, 253, 136, 20
8, 247, 238
540 DATA 121, 128, 208, 181, 96
550 DATA 162, 16, 189, 240, 3, 240,
12, 168
560 DATA 136, 240, 12, 233, 27, 16,
4, 152
570 DATA 157, 240, 3, 232, 208, 236,
 96, 169
580 DATA 26, 41, 255, 208, 243, 162,
 16, 189
590 DATA 240, 3, 168, 24, 105, 0, 24
0, 13
600 DATA 233, 26, 16, 9, 192, 26, 16
, 9
610 DATA 200, 152, 157, 240, 3, 232,
 208, 231
620 DATA 96, 169, 1, 9, 0, 208, 243
```

RUTINAS C.M. DE CIFRADO

```
.,  807C A9 10    LDA #$10
.,  807E 8D 79 80 STA $8079
.,  8081 AE 79 80 LDX $8079
.,  8084 BD F0 03 LDA $03F0,X
.,  8087 A8       TAY
.,  8088 88       DEY
.,  8089 18       CLC
.,  808A 69 00    ADC #$00
.,  808C 30 2D    BMI $80BB
.,  808E 18       CLC
.,  808F E9 1A    SBC #$1A
.,  8091 10 28    BPL $80BB
.,  8093 B9 5F 80 LDA $805F,Y
.,  8096 48       PHA
.,  8097 29 C0    AND #$C0
.,  8099 F0 1C    BEQ $80B7
.,  809B 68       PLA
.,  809C 48       PHA
.,  809D 29 80    AND #$80
.,  809F D0 10    BNE $80B1
.,  80A1 20 1B 80 JSR $801B
.,  80A4 20 22 80 JSR $8022
.,  80A7 68       PLA
.,  80A8 18       CLC
.,  80A9 2A       ROL
.,  80AA 2A       ROL
.,  80AB 29 FC    AND #$FC
.,  80AD 48       PHA
.,  80AE B8       CLV
.,  80AF 50 E6    BVC $8097
.,  80B1 20 3F 80 JSR $803F
.,  80B4 B8       CLV
.,  80B5 50 ED    BVC $80A4
.,  80B7 20 46 80 JSR $8046
.,  80BA 68       PLA
.,  80BB AC 7A 80 LDY $807A
```

```
.,   80BE AE 7B 80 LDX $807B
.,   80C1 CA       DEX
.,   80C2 D0 FD    BNE $80C1
.,   80C4 88       DEY
.,   80C5 D0 F7    BNE $80BE
.,   80C7 EE 79 80 INC $8079
.,   80CA D0 B5    BNE $8081
.,   80CC 60       RTS
.,   80CD A2 10    LDX #$10
.,   80CF BD F0 03 LDA $03F0,X
.,   80D2 F0 0C    BEQ $80E0
.,   80D4 A8       TAY
.,   80D5 88       DEY
.,   80D6 F0 0C    BEQ $80E4
.,   80D8 E9 1B    SBC #$1B
.,   80DA 10 04    BPL $80E0
.,   80DC 98       TYA
.,   80DD 9D F0 03 STA $03F0,X
.,   80E0 E8       INX
.,   80E1 D0 EC    BNE $80CF
.,   80E3 60       RTS
.,   80E4 A9 1A    LDA #$1A
.,   80E6 29 FF    AND #$FF
.,   80E8 D0 F3    BNE $80DD
.,   80EA A2 10    LDX #$10
.,   80EC BD F0 03 LDA $03F0,X
.,   80EF A8       TAY
.,   80F0 18       CLC
.,   80F1 69 00    ADC #$00
.,   80F3 F0 0D    BEQ $8102
.,   80F5 E9 1A    SBC #$1A
.,   80F7 10 09    BPL $8102
.,   80F9 C0 1A    CPY #$1A
.,   80FB 10 09    BPL $8106
.,   80FD C8       INY
.,   80FE 98       TYA
.,   80FF 9D F0 03 STA $03F0,X
.,   8102 E8       INX
.,   8103 D0 E7    BNE $80EC
.,   8105 60       RTS
.,   8106 A9 01    LDA #$01
.,   8108 09 00    ORA #$00
.,   810A D0 F3    BNE $80FF
.,   810C 00       BRK
.,   810D FF       ???
```



Respetando las variables

Para no borrar las variables cuando hacemos RUN, nada más fácil que hacer GOTO a la primera línea.

A.K. RUN EE.UU.

Ahorrando tiempo en el C-64

Para no tener que esperar el tiempo que el ordenador emplea en mostrar el programa que ha encontrado cuando busca en la cinta, basta con pulsar la tecla COMMODORE.

B.B.B. RUN EE.UU.

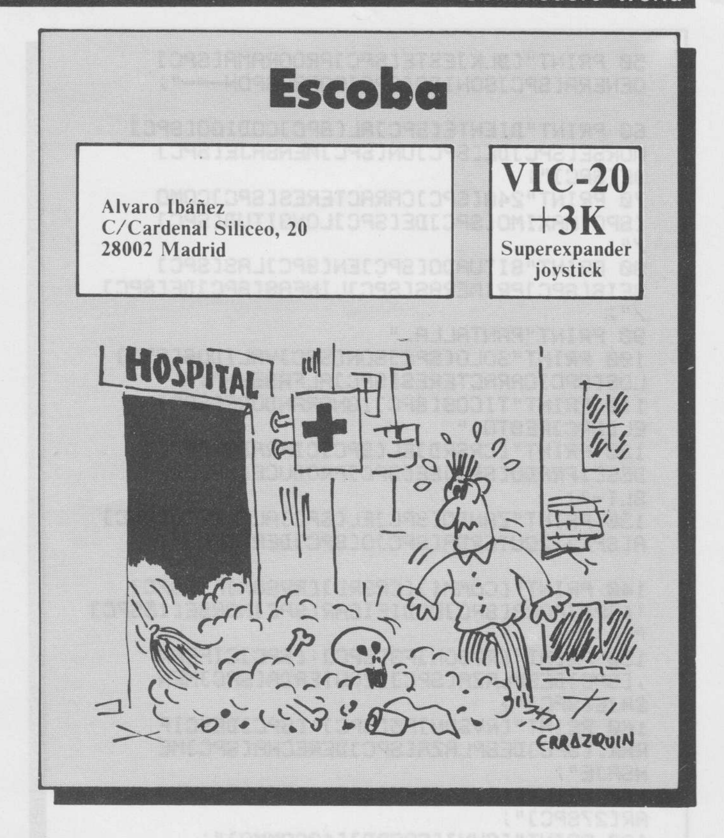
Scroll abajo

Para hacer un scroll abajo en el VIC-20, podemos utilizar la siguiente línea:

10 PRINT"[HOME][CRSR D][CRSR L][INSERT]":poke 218,158.

Cada vez que lo hagamos, toda la pantalla bajará una línea.

R.A.A. RUN EE.UU.



Escoba

Alvaro Ibáñez
C/Cardenal Siliceo, 20
28002 Madrid

VIC-20 +3K
Superexpander joystick

"La escoba" es un juego de cartas que he adaptado para que se pueda jugar contra el ordenador.

Para los que no sepáis cómo se juega os explicaré las reglas: A cada jugador se le dan tres cartas, y al principio se colocan cuatro sobre la mesa. Lo que hay que hacer es intentar sumar 15 con UNA de nuestras cartas y una o más de las que hay sobre la mesa. El valor de las cartas es el que indican; la J=8, la Q=9 y el rey (K)=10.

De modo que si por ejemplo en la mesa hay un 5, un 4 y un 2, y nosotros tenemos un rey, podemos llevarnos el 5, y si tenemos un caballo (Q), nos podemos llevar a la vez el 4 y el 2.

Si no queremos o no podemos sumar quince, hay que echar una carta.

Cuando se nos acaban las tres cartas, se reparten otras tres y así hasta que se acaban. Las últimas cartas se las lleva el que haya hecho la última jugada ganando puntos.

Lo que se valora al final de cada partida son:

Las cartas: Un punto para el que tenga más cartas.

Los oros: Para el que tenga más oros (los oros son las cartas que tienen un asterisco).

Los sietes: Para el que tenga más sietes.

El velo (el siete de oros): Un punto para el que lo tenga.

Escobas: Un punto por cada escoba que se haya hecho.

Una escoba se hace cuando al sumar quince no queda ninguna carta en la mesa.

El ganador es el que más puntos suma.

El programa

Para jugar tenemos que colocarnos sobre las cartas que queremos coger y pulsar el botón del joystick. Si no tienes joystick o superexpander puedes cambiar las líneas del programa que se indican a continuación del listado.

Cuando se han señalado todas las cartas que se van a coger, se pulsa el botón ENTER que está en la parte izquierda de la pantalla. Si nos equivocamos podemos pulsar el botón CLEAR. El programa detecta errores, de modo que no podemos hacerle trampas.

Líneas a cambiar si no se tiene joystick o superexpander:

```
1005 P=PEEK(197):IFP=64THEN1030
1010 IFP=32THEN1050
1015 POKEK,E:K=K+(P=17)—(P=18)+((P=52)—(P=21))*22
2050 IFPEEK(197)=64THEN2050
READY.
```

```
0 GOTO8:(C)AIB
3 T=SU:FORI=1TOLEN(S$):N=ASC(MID$(S$
,I,1))-64:P=0:IFMID$(R$,I,1)="*"THEN
P=1
5 T=T+1-(N=7)*SI+P*RO-((N=7)AND(P=1)
)*31:NEXT
6 T=T-(Q=0)*100-(Q<5)*2-(Q>14):IFT>B
ETHENBE=T:BE$=A$+S$+R$
7 RETURN
8 DIMC(39),J(2),V(2),D(9),A(10),B(10
):POKE36879,25:PRINT"[CLR][CRSRD][CRSRR]
[SPC]"
9 Y$="[HOM][22CRSRD]":X$="[22CRSRR]
10 C$="[SHIFTO][COMMT][SHIFTP][3CRSRL]
[CRSRD][COMMG][SPC][COMMM][3CRSRL][CRSRD]
[COMMG][SPC][COMMM][3CRSRL][CRSRD][COMMG]
[SPC][COMMM][3CRSRL][CRSRD][SHIFTL][COMM@]
[SHIFT@]":P$="*[SHIFTA]1/":N$="12345
67JQK":S=7680:CO=30720
11 D$="[COMMM][3COMM+][4CRSRL][CRSRD]
[COMMM][3COMM+][4CRSRL][CRSRD][COMMM]
[3COMM+][4CRSRL][CRSRD][COMMM][3COMM+]
[4CRSRL][CRSRD][COMMM][3COMM+]
15 PRINT"[HOM]"TAB(15)"VIC":PRINT:PR
INTTAB(15)"ESC=00":PRINTTAB(15)"PUN=
00[2CRSRL]"SPC(-(P2<10))MID$(STR$(P2
),2)
16 PRINTLEFT$(Y$,19)TAB(15)"TU":PRIN
T:PRINTTAB(15)"ESC=00":PRINTTAB(15)"
PUN=00[2CRSRL]";
17 PRINTSPC(-(P1<10))MID$(STR$(P1),2
)LEFT$(Y$,19)"[RVSON]E[SPC][2CRSRL][CRSRD]
[2SPC][2CRSRL][2CRSRD]C[SPC][2CRSRL]
[CRSRD][2SPC][HOM]
35 FORI=0TO39:C(I)=I:NEXT:FORI=0TO9:
D(I)=-1:NEXT
40 FORI=0TO39:B=INT(RND(0)*40):R=C(I
):C(I)=C(B):C(B)=R:NEXT
45 M=0:RO=3:SI=12:E1=0:E2=0:O1=0:O2=
0:S1=0:S2=0:FORI=1TO4:C=C(M):GOSUB87
5:M=M+1:NEXT
65 V1=0:V2=0:C1=0:C2=0:FORJU=1TO6:GO
SUB950:FORPL=1TO3:GOSUB1000:GOTO100
75 POKE7703,32:NEXT:NEXT
90 FORJ=0TO9:IFD(J)=-1THEN95
91 C=D(J):N=C:IFLA=2THENGOSUB550
92 IFLA=1THENGOSUB1152
93 Y=6-6*(J>4):X=1+(J-INT(J/5)*5)*4:
GOSUB975
95 NEXT:GOTO2000
100 TD=0:FORI=0TO9:IFD(I)=-1THEN104
103 TD=TD+D(I)-INT(D(I)/10)*10+1
104 NEXT
105 BE=0:BE$="":S$="":R$="":POKE7703
,160
110 B$="":O$="":FORI=0TO9
115 IFD(I)=-1THEN140
120 P=INT(D(I)/10)
125 N=D(I)-P*10
130 B$=B$+CHR$(65+N)
135 O$=O$+MID$(P$,P+1,1)
140 NEXT:L=LEN(B$)
141 IFL=0THEN400
145 IFL=1THEN185
150 FORI=1TOLEN(B$)-1
151 FORJ=1TOLEN(B$)-I
155 IFASC(MID$(B$,J,1))>ASC(MID$(B$,
J+1,1))THEN180
160 R$=MID$(B$,J,1):S$=MID$(B$,J+1,1
)
165 B$=LEFT$(B$,J-1)+S$+R$+MID$(B$,J
+2)
170 R$=MID$(O$,J,1):S$=MID$(O$,J+1,1
)
175 O$=LEFT$(O$,J-1)+S$+R$+MID$(O$,J
+2)
180 NEXT:NEXT
185 FORJ=0TO2:S$="":R$=""
191 IFV(J)=-1THEN300
195 P=INT(V(J)/10)
```

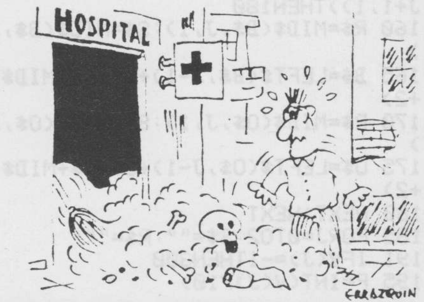
```
200 N=V(J)-P*10
201 A$=CHR$(65+N)+MID$(P$,P+1,1)
205 Q=TD+N-14:SU=1-(P=0)*RO-(N=6)*SI
-(V(J)=6)*31
210 X(1)=14-N:A(1)=1:B=1
215 N=ASC(MID$(B$,A(B),1))-64
220 IFN>X(B)THEN280
230 IFN<X(B)THEN310
240 S$="":R$="":FORI=1TOB
250 S$=S$+MID$(B$,A(I),1)
255 R$=R$+MID$(O$,A(I),1)
260 NEXT
270 GOSUB3
280 IFA(B)<>LTHEN330
290 IFB>1THENB=B-1:GOTO330
300 NEXT:GOTO350
310 IFA(B)=LTHEN290
320 A(B+1)=A(B):X(B+1)=X(B)-N:B=B+1
330 A(B)=A(B)+1:GOTO215
350 IFBE$=""THEN400
355 GOSUB800
360 GOTO500
400 BE=0:FORJ=0TO2:IFV(J)=-1THEN430
410 P=INT(V(J)/10)
415 N=V(J)-P*10
420 V=-(N+TD<4)*10-(P<>0)*3-(N<>6)*5
-((N<>6)AND(P<>0))-((N+TD)>13)*5-((N
+TD)<>7)
425 IFV>=BETHENBE=V:BE$=CHR$(65+N)+M
ID$(P$,P+1,1)
430 NEXT
435 N=ASC(LEFT$(BE$,1))-65:FORI=0TO3
:IFRIGHT$(BE$,1)<>MID$(P$,I+1,1)THEN
NEXT
436 P=I:C=N+P*10
437 FORI=0TO2:IFV(I)=CTHENV(I)=-1:J=
I
438 NEXT
440 Y=0:X=3+J*4:POKES+X+Y*22,32:GOSU
B900:FORT=1TO999:NEXT:Y=1:GOSUB975
450 GOSUB875
455 GOTO75
500 N=ASC(LEFT$(BE$,1))-65:FORI=0TO3
:IFMID$(BE$,2,1)<>MID$(P$,I+1,1)THEN
NEXT
505 C=N+I*10:GOSUB550
515 FORI=3TO2+L
520 N=ASC(MID$(BE$,I,1))-65:FORJ=0TO
3:IFMID$(BE$,I+L,1)=MID$(P$,J+1,1)TH
ENP=J
525 NEXTJ:C=N+P*10
530 GOSUB550
540 NEXTI
543 T=0:FORH=0TO9:T=T+D(H):NEXT:IFT=
-10THENE2=E2+1
544 PRINT"[HOM][2CRSRD]"SPC(19-(E2<1
0))MID$(STR$(E2),2)
545 LA=2:GOTO75
550 IFC<10THENO2=O2+1:IFO2>5THENRO=0

555 IFC-INT(C/10)*10=6THENS2=S2+1:IF
S2>2THENSI=0
560 IFC=6THENV2=1
565 C2=C2+1
575 RETURN
800 N=ASC(LEFT$(BE$,1))-65:FORI=0TO3
:IFMID$(BE$,2,1)<>MID$(P$,I+1,1)THEN
NEXT
810 P=I
815 C=N+P*10
820 FORI=0TO2:IFV(I)<>CTHENNEXT
825 V(I)=-1
830 Y=0:X=3+I*4:POKES+X+Y*22,32:GOSU
B900
835 L=(LEN(BE$)-2)/2
840 FORT=1TO1000:NEXT:FORH=3TOL+2
841 N=ASC(MID$(BE$,H,1))-65:FORJ=0TO
3:IFMID$(BE$,H+L,1)<>MID$(P$,J+1,1)T
HENNEXT
842 P=J:C=N+P*10
```

```
843 FORJ=0TO9:IFD(J)<>CTHENNEXT
844 D(J)=-1:Y=6-6*(J>4):X=1+(J-INT(J
/5)*5)*4:GOSUB975
845 J=10:NEXTJ:NEXTH:FORI=0TO2:IFPEE
K(S+3+I*4)<>79THEN847
846 X=3+I*4
847 NEXTI:Y=1:GOSUB975
849 RETURN
850 PRINTLEFT$(Y$,19)SPC(18);:FORI=1
TO4:PRINT"[RVSON]E[RVSOFF]";:FORT=1T
O200:NEXT:PRINT"[CRSRL][SPC][CRSRL]"
;
855 FORT=1TO200:NEXT:NEXT:RETURN
875 X=INT(RND(1)*5)*4+1:Y=6-6*(RND(1
)>.5):IFPEEK(S+X+Y*22)<>32THEN875
900 P=INT(C/10)
902 N=C-P*10
903 PRINTLEFT$(Y$,Y+1)LEFT$(X$,X)C$"
[HOM]
904 PRINTLEFT$(Y$,Y+2)LEFT$(X$,X+1)M
ID$(P$,P+1,1)"[CRSRL][2CRSRD]"MID$(N
$,N+1,1)"[HOM]
905 IFY<18ANDY>5THEND(INT((X-1)/4)-5
*(Y=12))=C
906 RETURN
950 V(0)=C(M):V(1)=C(M+1):V(2)=C(M+2
):M=M+3
955 PRINT"[HOM][2CRSRR]";:FORI=1TO3:
PRINTD$"[5CRSRU]";:FORT=1TO275:NEXT:
NEXT
960 J(0)=C(M):J(1)=C(M+1):J(2)=C(M+2
):M=M+3
965 FORI=3TO11STEP4:X=I:Y=18:C=J(I/4
):GOSUB900:FORT=1TO200:NEXT:NEXT
970 RETURN
975 FORI=YTOY+4+(Y=1):PRINTLEFT$(Y$,
I)LEFT$(X$,X-1)"[4SPC][4CRSRL][CRSRD]
[SPC][3COMM@][HOM]":FORT=1TO99:NEXT:
NEXT
980 PRINTLEFT$(Y$,Y+5+(Y=1))LEFT$(X$
,X-1)"[4SPC][HOM]
985 RETURN
990 PRINTLEFT$(Y$,Y+1)LEFT$(X$,X-1)"
[COMMM][CRSRL][CRSRD][COMMM][CRSRL][CRSRD]
[COMMM][CRSRL][CRSRD][COMMM][CRSRL][CRSRD]
[COMMM][HOM]":RETURN
995 PRINTLEFT$(Y$,Y+1)LEFT$(X$,X-1)"
[SPC][CRSRL][CRSRD][SPC][CRSRL][CRSRD]
[SPC][CRSRL][CRSRD][SPC][CRSRL][CRSRD]
[SPC][HOM]":RETURN
1000 SP=0:CE=0:T$="":K=8094:GOTO1020

1005 P=RJOY(1):IFP=0THEN1030
1010 IF(PAND128)=128THEN1050
1015 POKEK,E:K=K+((PAND4)=4)-((PAND8
)=8)+(((PAND1)=1)-((PAND2)=2))*22
1016 IFK<7680THENK=K+506
1017 IFK>8185THENK=K-506
1020 E=PEEK(K):POKEK,43:POKEK+CO,6:G
OTO1005
1030 H=H+1:IFH<10THEN1005
1035 H=0:IFCK=1THENCK=0:POKEK,E:GOTO
1005
1040 CK=1:POKEK,43:GOTO1005
1050 POKEK,E:H=9:CK=0
1051 KY=INT((K-7680)/22):KX=(K-7680)
-22*KY
1052 IFKY<6ORE=32THEN1015
1053 IFKY>17ANDKX>14THEN1015
1055 IFE>128ANDKY<20THEN1100
1056 IFE>128THEN1095
1060 IFKY>17THEN1085
1065 CA=INT(KX/4):Y=6:IFKY>11THENCA=
CA+5:Y=12
1070 X=1+(CA-INT(CA/5)*5)*4:IFPEEK(S
+X+Y*22-1)<>32THEN1015
1075 T$=T$+CHR$(48+CA):GOSUB990:GOTO
1015
1085 IFCE=1THEN1015
1086 CE=1:CA=INT((KX-3)/4):T$=CHR$(4
8+CA)+T$
1087 Y=18:X=3+CA*4:GOSUB990:GOTO1015

1090 IFCE=0ORLEN(T$)=1THEN1095
1091 IFSP=1THENIFCE=0ORLEN(T$)<>1THE
N1095
1092 IFSP=1THENX=INT(RND(1)*5)*4+1:Y
=6-6*(RND(1)>.5):C=J(VAL(T$)):GOSUB9
00:GOTO2000
1093 T=VAL(LEFT$(T$,1)):FORI=2TOLEN(
T$):T=T+VAL(MID$(T$,I,1)):NEXT:IFT<>
15THEN1095
1095 IFT$=""THEN1015
1096 IFCE=1THENCE=0:Y=18:X=3+VAL(LEF
T$(T$,1))*4:GOSUB995:T$=MID$(T$,2):
1097 FORI=1TOLEN(T$):Y=6:V=VAL(MID$(
T$,I,1)):IFV>4THENY=12
1098 X=1+(V-INT(V/5)*5)*4:GOSUB995:N
EXT:T$="":SP=0:GOTO1015
1100 L=LEN(T$):IFL=0THEN1015
1105 IFL=1ANDCE=0THENGOSUB850:GOTO10
95
1110 IFL=1THENY=18:X=3+VAL(T$)*4:GOS
UB975:C=J(VAL(T$)):GOSUB875:RETURN
1115 IFCE=0THENGOSUB850:GOTO1095
1120 V=J(VAL(LEFT$(T$,1))):T=V-INT(V
/10)*10+1:FORI=2TOLEN(T$):V=D(VAL(MI
D$(T$,I,1)))
1125 T=T+V-INT(V/10)*10+1:NEXT
1130 IFT<>15THENGOSUB850:GOTO1095
1135 Y=18:X=3+VAL(LEFT$(T$,1))*4:GOS
UB975
1140 FORJ=2TOLEN(T$):N=VAL(MID$(T$,J
,1)):Y=6-6*(N>4):X=1+(N-INT(N/5)*5)*
4:GOSUB975
1145 NEXT
1150 FORI=2TOLEN(T$):N=D(VAL(MID$(T$
,I,1))):D(VAL(MID$(T$,I,1)))=-1:GOSU
B1152:NEXT
1151 N=J(VAL(LEFT$(T$,1))):GOSUB1152
:GOTO1157
1152 IFN<10THENO1=O1+1
1153 IFN-INT(N/10)*10=6THENS1=S1+1
1154 IFN=6THENV1=1
1155 C1=C1+1
1156 RETURN
1157 FORI=0TO9:IFD(I)=-1THENNEXT:E1=
E1+1
1159 PRINTLEFT$(Y$,21)SPC(19-(E1<10)
)MID$(STR$(E1),2)
1160 LA=1:RETURN
2000 PRINT"[HOM][7CRSRD][7SPC]TU","[SPC]
VIC
2015 PRINT:PRINT"OROS[2SPC]"O1,O2:PR
INT"SIETES"S1,S2:PRINT"CARTAS"C1,C2:
PRINT"VELO[2SPC]"V1,V2:PRINT"ESCOBA"
E1,E2
2041 T1=-(O1>O2)-(S1>S2)-(C1>C2)-(V1
=1)+E1
2042 T2=-(O2>O1)-(S2>S1)-(C2>C1)-(V2
=1)+E2
2043 P1=P1+T1:P2=P2+T2:PRINT:PRINT"[SPC]
TOTAL"T1,T2
2050 IF(RJOY(0)AND128)<>128THEN2050
2055 FORI=7790TO8054:POKEI,32:NEXT:G
OTO15
```



Omnibus

Rafael Alberto García Medina
Carretera de Granada, nº 4, 3º C
Guadix (Granada)

C-64



Es un juego fascinante en el que la astucia y habilidad del jugador cuenta mucho.
Se puede jugar desde el teclado o con el Joystick en el Port 2.
Una nave intergaláctica llamada OMNIBUS es atacada por 99 misiles enemigos. Tu misión es la de tripular el OMNIBUS y esquivar el ataque enemigo.
Se gana cuando se lleva la nave hasta la parte inferior de la pantalla o cuando se esquivan los 99 misiles.
Se pierde cuando el OMNIBUS es alcanzado por algún misil.
Al final del juego se visualiza los puntos obtenidos y los centones (es decir el tiempo) realizado durante el vuelo.

```
1 X=1:Y=1:DX=1:DY=1:POKE53280,1:PRIN
T"[RED][CLR]":POKE53281,1 :FORK=1TO3
13
2 POKE1024+X+40*Y,78:X=X+DX:IFX=0ORX
=39THENDX=-DX
3 Y=Y+DY:IFY=0ORY=24THENDY=-DY
4 NEXTK:PRINTTAB(16)"[CRSRD][BLK][RVSON]
OMNIBUS":PRINTTAB(7)"[5CRSRD][SPC]DE
[SPC]RAFAEL[SPC]GARCIA[SPC]MEDINA[SPC]
"
5 FORK=1TO4000:NEXTK
6 PRINT"[3CRSRD]SE[SPC]PUEDE[SPC]JUG
AR[SPC]DESDE[SPC]EL[SPC]TECLADO[SPC]
O[SPC]CON[SPC]EL[SPC]JOISTIK[SPC]EN[SPC]
EL[SPC]PORT[SPC]2"
7 PRINTTAB(3)"F.....DERECHA"
8 PRINTTAB(3)"J.....IZQUIERDA"
9 PRINTTAB(3)"B.....ABAJO"
10 PRINTTAB(3)"Y.....RETORNA[SPC]ARR
IBA"
11 PRINT"[RVSON][3CRSRD]VAMOS[SPC]A[SPC]
COMENZAR[SPC]EL[SPC]VIAJE[SPC]INTERG
ALACTICO":PRINT"[RVSON]PULSA[SPC]UNA
[SPC]TECLA."
12 GETW$:IFW$=""THEN12
13 TI$="000000":PRINT"[CLR][10CRSRD]
":POKE53269,0
14 X=130:Y=50
15 A=INT(TI/60):IFA*70=>6930THEN5100

20 POKE53280,T:POKE53281,T+1
30 T=INT(RND(0)*30)
1000 RESTORE
1020 V=53248:POKEV+34,3
1030 POKE53269,4:POKE2042,13
1035 IFY<20THENY=20
1036 IFY=>260THEN5100
1037 IFX=>250THENX=250
1038 IFX=<10THENX=10
1040 FORN=0TO62:READQ:POKE832+N,Q:NE
XT
1045 X=X:Y=Y:POKE53252,X:POKE53253,Y

1050 DATA0,0,0,0,0,0,0,0,0,1,1,0,1,1
31,0,1,131,0,1,199,0,1,199,0,1,199,0
,1,199
1060 DATA0,0,238,0,0,254,0,0,16,0,0,
124,0,0,16,0,0,56,0,1,255,0,0,56,0,0
,16,0
1070 DATA0,16,0,0,16,0
2000 POKE53277,PEEK(53277)OR(2↑2):PO
KE53271,PEEK(53271)OR(2↑2)
3000 IFPEEK(V+31)ANDX=XTHEN5999
3500 GETR$:IFR$="F"THENX=X-30
3510 IFR$="J"THENX=X+30
3520 IFR$="B"THENY=Y+15
3530 IFR$="Y"THENY=Y-100
4000 JV=PEEK(56320)
4010 IF(JVAND1)=0THEN Y= Y+100*(Y>1)

4020 IF(JVAND2)=0THEN Y= Y-15*(Y<255
)
4030 IF(JVAND4)=0THEN X= X+30*(Y>1)
4040 IF(JVAND8)=0THEN X= X-30*(X<350
)
4100 PRINTTAB(T) "[RED][2SPC][SHIFTN]
[SHIFTM]
4110 PRINTTAB(T) "[2SPC][2COMM+]
4120 PRINTTAB(T) "[2SPC][2COMM+]
4130 PRINTTAB(T) "[2SPC][COMMG][COMMM]

4140 PRINTTAB(T) "[SPC][SHIFTN][COMMF]
[COMMD][SHIFTM]
4145 PRINTTAB(T) "[SPC]"A
4150 PRINTTAB(T) "[SPC][COMMG][2SHIFTB]
[COMMM][3SPC][8SPC]
4151 PRINTTAB(T) "[2SPC]**[3SPC]";
4152 PRINT
5000 FORL=54272TO54295:POKEL,0:NEXT
5010 POKE54296,15
5020 POKE54277,28.9
5040 POKE54273,93:POKE54272,2
5050 POKE54276,129
5060 POKE54276,16
5070 GOTO15
5100 PRINTTAB(4)"[RVSON][9SPC]MUY[SPC]
BIEN[SPC]CAPITAN!![SPC]"
5110 PRINTTAB(4)"[2CRSRD]LA[SPC]NAVE
[SPC]HA[SPC]SIDO[SPC]TRIPULADA"
5111 PRINTTAB(4)"PERFECTAMENTE"
5120 PRINT"[2CRSRD]"
5130 PRINTTAB(15)"[RVSON][8SPC]"
5140 PRINTTAB(16)"[RVSON][2SPC]O.K![2SPC]
"
5150 PRINTTAB(17)"[RVSON][8SPC]"
5999 POKEV+28,4:POKEV+38,3:POKEV+37,
0
6000 FORB=1TO16
6010 POKE53280,B:POKE53281,B+3
6020 NEXTB
7000 POKE53281,7
8000 FORL=54272TO54295:POKEL,0:NEXT
8010 POKE54296,15
8020 POKE54277,28.9
8040 POKE54273,11:POKE54272,1
8050 POKE54276,129
8070 FORD=1TO3000:NEXT
9000 POKE53276,PEEK(53276)AND(255-2↑
2)
10005 POKE53281,7:POKEV+21,0
10010 PRINT"[4CRSRD]GAME[SPC]OVER"
```

```
10020 PRINT"TU[SPC]PUNTUACION[SPC]HA
[SPC]SIDO[SPC][RVSON]";A*70"[RVSOFF]
PUNTOS"
10025 PRINT"TIEMPO[SPC]DE[SPC]VUELO[SPC]
[RVSON]";A"[RVSOFF]CENTONES"
11000 PRINT"QUIERES[SPC]JUGAR[SPC]OT
RA[SPC]VEZ(S/N)"
11010 GETA$:IFA$="S"THEN13
11020 IFA$<>"N"THEN11010
```

Tron

Alvaro Ibáñez
Cardenal Siliceo, 20
28002 Madrid

VIC-20 sin expansión



Los que hayais visto la película TRON de Walt Disney, seguramente os habréis quedado asombrados por las fabulosas carreras de motos que tenían lugar entre los protagonistas. Este programa es una recreación de una de esas carreras de motos, en la que hemos de luchar contra el ordenador.
El objetivo es acorralar al ordenador, para que no tenga más remedio que chocarse, y, al mismo tiempo, evitar que el ordenador nos haga lo mismo a nosotros.
Nuestra moto se controla con las teclas A,D,O y L, tal y como se indica en las instrucciones al principio del programa.
La barra de espacios la podemos usar en caso de que nos veamos acorralados y veamos que nos chocamos sin remedio. Entonces nos transportará a otro lugar de la pantalla. Tiene el inconveniente de que a veces aparecemos en un lugar que ya está ocupado, y morimos irremisiblemente.
Hay que tener en cuenta que el ordenador también puede hacer uso del hiperespacio, por lo que es realmente difícil ganarle.
Si queremos que el ordenador no haga uso de él, basta con cambiar la línea 750 por: 750 REM, y si queremos que nosotros tampoco podamos usarlo hacemos lo mismo con la línea 250.
El ganador es el que antes llegue a 10 puntos.

```
5 REM(C)ALVARO IBANEZ
10 GOTO1000
40 D(1)=1:D(2)=-1:D(3)=-22:D(4)=22
45 BL=0:RO=0:R$="[HOM][WHT][3CRSRR]V
IC":B$="[HOM][WHT][11CRSRR]TU"
50 H=0:W=0
60 PRINT"[CLR]"
100 M=INT(RND(1)*484)+7702
105 K=INT(RND(1)*484)+7702:IFK=MTHEN
105
110 POKEM,160:POKEK+30720,4:POKEK,16
0
115 PRINTR$;RO:PRINTB$;BL
120 GETA$:IFA$<>""THEN120
125 GETA$:IFA$=""THEN125
170 GOTO210
200 GETA$
210 IFA$="A"THEND=-1
220 IFA$="D"THEND=1
230 IFA$="O"THEND=-22
240 IFA$="L"THEND=22
250 IFA$="[SPC]"THENM=INT(RND(1)*484
)+7702
260 M=M+D
265 IFPEEK(M)=160THENG=1:GOTO800
270 IFM<7702THENM=M+484:GOTO280
275 IFM>8185THENM=M-484:GOTO280
280 IFPEEK(M)=160THENG=1:GOTO800
290 POKEM,160
300 IFH=1THEN400
310 H=1:X=INT(RND(1)*10)+1:J=INT(RND
(1)*4)+1
320 IFD(J)<>-PTHEN390
335 ONJGOTO340,350,360,370
340 J=INT(RND(1)*3)+2:GOTO390
350 J=INT(RND(1)*3)+2:IFJ=2THENJ=1
355 GOTO390
360 J=INT(RND(1)*3)+1:IFJ=3THENJ=4
365 GOTO390
370 J=INT(RND(1)*3)+1
390 P=D(J)
400 IFPEEK(K+P)=160THEN500
403 K=K+P
405 IFK<7702THENK=K+484
410 IFK>8185THENK=K-484
414 IFPEEK(K)=160THEN750
415 POKEK+30720,4:POKEK,160
420 W=W+1:IFW>XTHENH=0:W=0
430 GOTO200
500 IFP=-1THENA1=-22:A2=22:A3=1
510 IFP=1THENA1=-22:A2=22:A3=-1
520 IFP=-22THENA1=-1:A2=1:A3=22
530 IFP=22THENA1=-1:A2=1:A3=-22
550 IFPEEK(K+A1)=160ANDPEEK(K+A2)=16
0ANDPEEK(K+A3)=160THEN750
555 IFPEEK(K+A1)=160ANDPEEK(K+A2)=16
0THENP=A3:GOTO403
560 IFPEEK(K+A1)=160ANDPEEK(K+A3)=16
0THENP=A2:GOTO403
565 IFPEEK(K+A1)=160THENP=A3:GOTO403
570 IFPEEK(K+A2)=160ANDPEEK(K+A3)=16
0THENP=A1:GOTO403
575 IFPEEK(K+A2)=160THENP=A3:GOTO403
580 S=RND(1):IFS>.5THENP=A1:GOTO403
590 P=A2:GOTO403
750 K=INT(RND(1)*484)+7702
755 IFPEEK(K)=160THEN800
760 GOTO200
800 GOTO900
810 IFG=1THENRO=RO+1:G=0:GOTO830
820 BL=BL+1
830 IFRO<10ANDBL<10THENH=0:GOTO50
835 PRINT"[HOM][CRSRD]";:FORI=1TO22:
PRINT"[22SPC]";:FORT=1TO100:NEXT:NEX
T
840 A$="[HOM][10CRSRD][6CRSRR]GAME[SPC]
OVER"
845 FORI=1TOLEN(A$):PRINTMID$(A$,I,1
);:FORT=1TO70:NEXT:NEXT
850 IFRO=10THEN870
860 FORI=1TO10:PRINT"[HOM][22SPC]":F
ORT=1TO100:NEXT
```

```
865 PRINTB$;BL:FORT=1TO300:NEXT:NEXT
:GOTO1000
870 FORI=1TO10:PRINT"[HOM][22SPC]":F
ORT=1TO100:NEXT
875 PRINTR$;RO:FORT=1TO300:NEXT:NEXT
:GOTO1000
900 IFG=1THENY=M
910 IFG=0THENY=K
914 FORW=1TO06:POKEY,32:FORT=1TO60:N
EXT:POKEY,160:FORT=1TO90:NEXT:NEXT
915 Q1=Y-22:Q2=Y-21:Q3=Y+1:Q4=Y+23:Q
5=Y+22:Q6=Y+21:Q7=Y-1:Q8=Y-23
917 FORE=1TO15
920 POKEQ1,93:POKEQ2,78:POKEQ3,67:PO
KEQ4,77:POKEQ5,93:POKEQ6,78:POKEQ7,6
7:POKEQ8,77
930 Q1=Q1-22:Q2=Q2-21:Q3=Q3+1:Q4=Q4+
23:Q5=Q5+22:Q6=Q6+21:Q7=Q7-1:Q8=Q8-2
3
940 POKEQ1+66,32:POKEQ2+63,32:POKEQ3
-3,32:POKEQ4-69,32
950 POKEQ5-66,32:POKEQ6-63,32:POKEQ7
+3,32:POKEQ8+69,32
960 NEXT:GOTO810
1000 PRINT"[CLR]";:POKE36879,110
1010 PRINT"[BLU][RVSON][9SPC]TRON[9SPC]
[WHT]":FORI=38400TO38421:POKEI,1:FOR
T=1TO90:NEXT:NEXT
1015 PRINT"[CRSRD][SPC](C)[SPC]ALVAR
O[SPC]IBAN[CRSRU][CRSRL][SHIFTR][CRSRD]
EZ"
1020 PRINT"[CRSRD][8SPC]MANDOS"
1025 PRINT"[7SPC][8SHIFTE]"
1030 PRINT"[2CRSRD][3SPC]IZQ.-[RVSON]
A[RVSOFF][2SPC]ARRIBA-[RVSON]O[RVSOFF]
"
1040 PRINT"[CRSRD][2SPC]DCHA.-[RVSON]
D[RVSOFF][2SPC]ABAJO[SPC]-[RVSON]L[RVSOFF]
"
1050 PRINT"[2CRSRD][4SPC]PARA[SPC]SA
LTAR[SPC]AL[SPC]":PRINT"[CRSRD][3SPC]
HIPERESPACIO,[SPC]USA"
1055 PRINT"[CRSRD][6SPC][RVSON][2SPC]
SPACE[2SPC][RVSOFF]"
1060 PRINT"[2CRSRD][6SPC][GRN][SPC]P
ULSA[SPC][RVSON]F1[RVSOFF]"
1070 GETA$:IFA$<>"[F1]"THEN1070
1080 GOTO40
```



Números aleatorios

Para conseguir un número aleatorio entre 0 y 255:
x=int(rnd(1)*256)

Sin embargo, si quieres un número entre 0 y 15, por ejemplo:
PRINT PEEK(162) AND 15

Algunas combinaciones funcionan mejor que otras. ¡Pruébalo!

I.A. RUN EE.UU.

Caza de errores

Cuando el ordenador esté leyendo datas, y de pronto te dé un ILLEGAL QUANTITY ERROR, en vez de buscar por todas las líneas que contienen datas, teclea:

Print peek(64)*255+peek(63)

Esto dará el número de línea en la que está el error.

A.B. RUN EE.UU.

Basket

José y Eduardo Sánchez de Rojas
C/Peñuelas, 12, 5º-I
28005 Madrid

C-64

¡ES FANTASTICO!.. SOLO TIENE QUE APRENDER A SOLTAR EL BALON

Se trata de elegir un ángulo y una velocidad apropiadas para conseguir encestar. Sólo se dispone de diez fallos y la canasta cambia de posición cada vez que se encesta.
Las teclas de función controlan la velocidad y el ángulo de la pelota, mientras la tecla "A" pone el balón en movimiento.

10	Poke teclas repetitivas.
20	Define variables.
30	Color pantalla.
60-110	Define sprites.
120-160	Control velocidad y ángulo.
180	Posición inicial sprites. "A" para mover balón.
190-210	Lanzamiento balón.
220-240	Límites balón.
250	Posición balón.
260	Comprueba si hay enceste.
280-310	Imprime velocidad, ángulo y lanzamientos fallados.
320-420	Imprime puntuación y define nueva posición de canasta.
420-260	Caída balón.
470-530	Decoración pantalla.
540-580	Pregunta por nuevo juego.
590-640	Datas sprites.

```
0 REM****************
1 REM* J,E SANCHEZ  *
2 REM*   BASKET     *
3 REM****************
10 POKE650,255
20 X=50:Y=170:XC=225:YC=130:VO=0:A=0
:P=0:LZ=0
30 POKE53281,0:POKE53280,0
40 PRINT"[CLR]"
50 GOSUB470
60 V=53248
70 POKEV+21,6
80 POKE2042,192:POKE2041,13:POKEV+23
,2:POKEV+29,2:POKEV+41,11:POKEV+40,1

90 FORN=0TO62:READQ1:POKE12288+N,Q1:
NEXT
100 FORN=0TO62:READQ2
110 POKE832+N,Q2:NEXT
```

```
120 GETH$
130 IFH$="[F3]"THENVO=VO-1
140 IFH$="[F5]"THENA=A+1
150 IFH$="[F7]"THENA=A-1
160 IFH$="[F1]"THENVO=VO+1
170 GOSUB280
180 POKEV+2,XC:POKEV+3,YC:POKEV+4,50
:POKEV+5,170:IFH$<>"A"THEN120
190 TI$="000000":LZ=LZ+1
200 Y =INT(-(VO*SIN(A*π/180)*(TI/100
)-4.9*(TI/100)↑2)+170)
210 X =INT((VO*COS(A*π/180)*(TI/100)
)+50)
220 IFY<0THEN420
230 IFX>255ORX<0THEN420
240 IFY>227THENPOKEV+4,X:POKEV+5,227
:GOTO120
250 POKEV+4,X:POKEV+5,Y
260 IFX>XCANDX<XC+20ANDY>YC-20ANDY<Y
C-15THEN320
270 GOTO200
280 PRINT"[HOM]VELOCIDAD=";VO
290 PRINT"[HOM][2CRSRD]LANZAMIENTO="
LZ:IFLZ=10THEN540
300 PRINT"[HOM][CRSRD]ANGULO=";A
310 RETURN
320 P=P+2:LZ=LZ-1:FORD=1054TO1062:PO
KED,121:POKED+54272,2:NEXT
330 FORD=1174TO1182:POKED,120:POKED+
54272,2:NEXT
340 FORD=1093TO1253STEP40:POKED,118:
POKED+54272,2:NEXT
350 FORD=1103TO1302STEP40:POKED,117:
POKED+54272,2:NEXT
360 FORD=1294TO1302:POKED,120:POKED+
54272,2:NEXT
370 XC=INT(RND(1)*41)+150
380 YC=INT(RND(1)*16)+125
390 PRINTTAB(30)"[RVSON]!CANASTA!"
400 PRINTTAB(30)"[2CRSRU][RVSON][9SPC]
":FORT=0TO1100:NEXT
410 PRINTTAB(30)"[RVSON][9SPC]":PRIN
TTAB(33)"[CRSRD]"P
420 FORY=YTO227:IFX>255THENX=255
430 IFY<0THENY=0
440 POKEV+4,X:POKEV+5,Y
450 NEXT
460 GOTO120
470 FORJ=0TO39:POKE1984+J,121:POKE56
256+J,7:NEXT
480 FORI=0TO159:POKE1144+I,81:POKE55
416+I,8:IFI>28ANDI<40THENPOKE1144+I,
32
490 IFI>68ANDI<80THENPOKE1144+I,32
500 IFI>108ANDI<120THENPOKE1144+I,32

510 IFI>148ANDI<160THENPOKE1144+I,32

520 NEXT
530 RETURN
540 PRINT"[CLR][CRSRD][RVSON]HAS[SPC]
CONSEGUIDO[RVSOFF][SPC]";P;"PUNTOS"
550 PRINT"QUIERES[SPC]VOLVER[SPC]A[SPC]
JUGAR?(S/N)"
560 GETV$:IFV$=""THEN560
570 IFV$="S"THEN20
580 END
590 DATA0,254,0,3,239,128,7,239,192,
27,239,176,61,239,120,126,238,252,12
6
600 DATA238,252,254,108,254,255,109,
254,255,109,254,255,109,254,255,109,
254
610 DATA254,108,254,254,238,254,126,
238,252,125,239,124,59,239,184,23,23
9,208
620 DATA15,239,224,3,239,128,0,254,0

630 DATA0,255,0,3,0,192,12,0,63,8,0,
48,12,0,63,7,0,192,6,255,64,5,17,64,
6,170
640 DATA192,2,68,128,2,170,128,3,17,
128,2,170,128,1,69,0,1,171,0,1,17,0
650 DATA,,,,,,,,,,,,,,,,,,,
```

Castle

Alvaro Ibáñez
Cardenal Siliceo, 20
28002 Madrid

VIC-20 sin expansión



Este juego es realmente interesante porque hace falta habilidad y sangre fría.
Los invasores están subiendo por las paredes de nuestro castillo, y nosotros hemos de evitarlo tirándoles piedras a la cabeza. Hemos de acercarnos a uno de los extremos de la pantalla en el que aprece una piedra. La cogemos, nos situamos encima del hombrecillo al que vamos a lanzarle la piedra y se la tiramos.
Si le damos en la cabeza lo derribaremos, y caerá al suelo. Si le damos en un brazo puede que se enfade y nos la devuelva. En este caso hemos de evitar que nos dé a nosotros, aunque también podemos cogerla al vuelo.
Si el invasor en su caída choca con alguno de sus compañeros, logrará sujetarse y continuar su escalada, pero el de abajo será el que sufra las consecuencias.
A medida que vamos sobreviviendo, aparecerán más y más invasores, hasta que no podamos contenerlos, lleguen arriba y nos maten. Entonces el juego comenzará de nuevo.
Nuestro hombrecillo se controla con las teclas Z,C para los lados y M para coger las piedras y tirarlas.
El inconveniente de estas teclas es que si las mantenemos pulsadas puede suceder que al coger la piedra la tiremos acto seguido. Para evitarlo hay que cambiar las siguientes líneas:

```
100 get a$: if a$="" then 190
110 if asc (a$)=13 then 500
130 m=m+(a$="(crsr 1)")—(a$="(crsr r)")
```

De este modo podremos movernos con las teclas del cursor y coger las piedras con la tecla RETURN.

```
1 PR=.95:GOTO17:(C)AIB
2 PRINT"[HOM][2CRSRD]"SPC(M)"[SPC]"
3 PRINTSPC(M)"[3SPC][3CRSRL][CRSRD][3SPC]
[3CRSRL][CRSRD][3COMM@]"
```

```
4 RETURN
5 PRINT"[HOM][2CRSRD]";
6 IFCTHEN10
7 PRINT
8 PRINTSPC(M)"[SPC]O[SPC][3CRSRL][CRSRD]
[SHIFTN][COMM+][SHIFTM][3CRSRL][CRSRD]
[SHIFT@][COMM@][SHIFTL]":RETURN
10 PRINTSPC(M)"[SPC][RVSON][SPC][RVSOFF]
[SPC]"
11 PRINTSPC(M)"[COMMM]O[COMMG][3CRSRL]
[CRSRD][SPC][COMM+][SPC][3CRSRL][CRSRD]
[SHIFT@][COMM@][SHIFTL]":RETURN
13 A$=RIGHT$(STR$(SC),LEN(STR$(SC))-
1)
14 PRINT"[HOM][7CRSRR]"SPC(5-LEN(A$)
)A$:RETURN
17 DIMX(8),Y(8),B$(3),K(8):CO=30720:
L=-1
18 PRINT"[CLR][4CRSRD]":FORI=0TO8:X(
I)=0:Y(I)=0:K(I)=0:NEXT:S=0:M=10:FOR
I=1TO22:PRINT"[COMM@]";:NEXT
30 FORI=7812TO8164STEP22:POKEI+CO,6:
POKEI+CO+21,6:POKEI,101:POKEI+21,103
:NEXT
50 FORI=8164TO8185:POKEI+CO,6:POKEI,
160:NEXT
60 B$(0)="[3SPC][3CRSRL][CRSRD][COMMM]
[SPC][COMMG][3CRSRL][CRSRD][SHIFTM]O
[SHIFTN][3CRSRL][CRSRD][SPC][RVSON][GRN]
[SPC][RVSOFF][BLU][SPC]"
65 B$(1)="[SPC][COMMG][SPC][3CRSRL][CRSRD]
[SPC]O[SHIFTN][3CRSRL][CRSRD][COMMM]
[RVSON][GRN][COMMG][RVSOFF][BLU][SPC]
[3CRSRL][CRSRD][SPC][COMMM][SPC]"
70 B$(2)="[SPC][COMMM][SPC][3CRSRL][CRSRD]
[SHIFTM]O[SPC][3CRSRL][CRSRD][SPC][RVSON]
[GRN][COMMG][RVSOFF][BLU][COMMG][3CRSRL]
[CRSRD][SPC][COMMG][SPC]"
75 B$(3)="[3SPC][3CRSRL][CRSRD][3SPC]
[3CRSRL][CRSRD][3SPC][3CRSRL][CRSRD]
[3SPC]"
77 Y$="[HOM][22CRSRD]"
80 X$="[23CRSRR]"
85 PRINT"[HOM][SPC]SCORE[SPC]00000":
GOSUB5
95 POKE7790,100:POKE7811,100
97 PL=1:IFRND(1)>.5THENPL=18
98 IFPL=1THENPOKE7790+CO,6:POKE7790,
160
99 IFPL=18THENPOKE7811+CO,6:POKE7811
,160
100 P=PEEK(197):IFP=64THEN190
110 IFP=36THEN500
120 GOSUB2
130 M=M+(P=33)-(P=34)
140 IFM<1THENM=1
150 IFM>18THENM=18
160 GOSUB5:GOSUB13
190 IFS=0THEN300
200 IFPEEK(S)=15THEN600
201 IFPEEK(S-1)=100ORPEEK(S+1)=100TH
ENPOKES,100:GOTO203
202 POKES,32
203 S=S+D:SC=SC+1:GOSUB13:IFPEEK(S)=
78THEN230
204 IFPEEK(S)=77THEN250
205 IFPEEK(S)=15THEN600
206 IFS>8163THENS=0:GOTO95
207 IFS<7702THENS=S+22:POKES,160:D=2
2
210 POKES+30720,6:POKES,160
220 GOTO300
230 IFS=7768+MTHENS=0:C=1:GOSUB5:GOT
O300
240 POKES,160:POKES+22,125:FORT=1TO2
00:NEXT
245 D=-22:S=S-22:POKES,160:POKES+22,
78:POKES+44,32:FORT=1TO99:NEXT:GOTO3
00
250 IFS=7770+MTHENS=0:C=1:GOSUB5:GOT
O300
260 POKES,160:POKES+22,109:FORT=1TO2
00:NEXT
265 D=-22:S=S-22:POKES,160:POKES+22,
77:POKES+44,32:FORT=1TO99:NEXT:GOTO3
00
300 L=L+1:IFL>8THENL=0
310 IFK(L)THEN350
320 IFRND(1)<PRTHEN100
330 Y(L)=19:X(L)=INT(RND(1)*18+1):K(
L)=1
340 IFPEEK(8075+X(L))<>32ORPEEK(8076
+X(L))<>32ORPEEK(8077+X(L))<>32THENK
(L)=0:GOTO100
345 GOTO353
350 PRINTLEFT$(Y$,Y(L));LEFT$(X$,X(L
));B$(3);"[HOM]"
352 Y(L)=Y(L)-1:IFY(L)=6THEN800
353 PRINTLEFT$(Y$,Y(L));
355 PRINTLEFT$(X$,X(L));
360 IFK(L)=1THENPRINTB$(1);:K(L)=2:G
OTO375
370 IFK(L)=2THENPRINTB$(2);:K(L)=1
375 PRINT"[HOM]"
380 GOTO100
500 IFM<>PLORC=1ORSTHEN550
510 GOSUB2
520 C=1:GOSUB5
525 POKE7790,100:POKE7811,100
530 GOTO190
550 IFC=0ORSTHEN190
555 PRINT"[HOM][2CRSRD]";
560 PRINTSPC(M)"[3SPC][3CRSRL][CRSRD]
[SPC][RVSON][SPC][RVSOFF][SPC][3CRSRL]
[CRSRD][SPC][COMM+][SPC][3CRSRL][CRSRD]
[SHIFT@][COMM@][SHIFTL]"
565 FORT=0TO100:NEXT
570 PRINT"[HOM][3CRSRD]";
575 PRINTSPC(M)"[COMMM]O[COMMG][3CRSRL]
[CRSRD][SPC][RVSON][SPC][RVSOFF][SPC]
[3CRSRL][CRSRD][SHIFT@][COMM@][SHIFTL]
"
580 FORT=0TO100:NEXT
581 PRINT"[HOM][3CRSRD]";
585 PRINTSPC(M)"[SPC]O[SPC][3CRSRL][CRSRD]
[COMMT][COMM+][COMMT][3CRSRL][CRSRD]
[SHIFT@][RVSON][SPC][RVSOFF][SHIFTL]
"
590 FORT=0TO100:NEXT
591 PRINT"[HOM][3CRSRD]";
595 PRINTSPC(M)"[SPC]O[SPC][3CRSRL][CRSRD]
[SHIFTN][COMM+][SHIFTM][3CRSRL][CRSRD]
[SHIFT@][COMM@][SHIFTL][2CRSRL][CRSRD]
[RVSON][SPC][RVSOFF]"
597 S=7813+M:C=0:D=22
599 GOTO190
600 FORI=0TO8:IFS=7681+Y(I)*22+X(I)T
HEN610
605 NEXT:GOTO830
610 R=I:PRINTLEFT$(Y$,Y(R));LEFT$(X$
,X(R));B$(0);"[HOM]"
620 IFPEEK(7681+66+Y(R)*22+X(R))<>32
ANDY(R)<18THEN670
630 Y(R)=Y(R)+1:S=S+22:SC=SC+10:GOSU
B13:IFY(R)>19THEN700
640 PRINTLEFT$(Y$,Y(R));
645 PRINTLEFT$(X$,X(R));
650 PRINTB$(0);"[HOM]"
660 GOTO620
670 S=S+88:IFPEEK(S)=15THEN690
675 IFPEEK(S-1)=15THENS=S-1:GOTO690
680 IFPEEK(S+1)=15THENS=S+1
690 POKES-20,32:POKES-21,32:POKES-22
,32:POKES-23,32:POKES-24,32
695 PRINTLEFT$(Y$,Y(R));LEFT$(X$,X(R
));"[2COMMM][SPC][3CRSRL][CRSRD][SPC]
O[SPC][3CRSRL][CRSRD][SPC][RVSON][GRN]
[SPC][RVSOFF][BLU][SPC][3CRSRL][CRSRD]
[2COMMM][SPC][HOM]"
```

```
699 SC=SC+500:GOSUB13:GOTO600
700 PRINTLEFT$(Y$,Y(R)-1);LEFT$(X$,X
(R));"[CRSRD][3SPC][3CRSRL][CRSRD][COMMM]
[SPC][COMMG][3CRSRL][CRSRD][SHIFTM]O
[SHIFTN]";"[HOM]":FORT=1TO70:NEXT
705 PRINTLEFT$(Y$,Y(R)-1);LEFT$(X$,X
(R));"[CRSRD][3SPC][3CRSRL][CRSRD][SHIFTM]
[SPC][SHIFTN][3CRSRL][CRSRD][SHIFTR]
[SHIFTW][SHIFTR]";"[HOM]":FORT=1TO70
:NEXT
706 PRINTLEFT$(Y$,19);LEFT$(X$,X(R))
;B$(3);"[HOM]"
710 K(R)=0:S=0:PR=PR-.01:SC=SC+100:G
OSUB13:GOTO95
800 PRINT"[HOM][5CRSRD]";LEFT$(X$,X(
L));"[COMM@]O[COMM@][3CRSRL][CRSRD][COMMZ]
[RVSON][GRN][SPC][RVSOFF][BLU][COMMX]
[3CRSRL][CRSRD][SHIFTN][SPC][SHIFTM]
":FORT=1TO300:NEXT
805 PRINT"[HOM][4CRSRD]";LEFT$(X$,X(
L));"[SPC]O[SPC][3CRSRL][CRSRD][SHIFT@]
[RVSON][GRN][SPC][RVSOFF][BLU][SHIFTL]
[3CRSRL][CRSRD][2COMMM][SPC][3CRSRL]
[CRSRD][3SPC]":FORT=1TO300:NEXT
810 PRINT"[HOM][3CRSRD]";LEFT$(X$,X(
L));"[SPC]O[SPC][3CRSRL][CRSRD][SHIFTN]
[RVSON][GRN][SPC][RVSOFF][BLU][SHIFTM]
[3CRSRL][CRSRD][SHIFT@][COMM@][SHIFTL]
[3CRSRL][CRSRD][3SPC]":FORT=1TO300:N
EXT
815 K=7769+X(L):D=(X(L)>M)-(X(L)<M):
K=K+D:POKEK-22,100:IFK=7769+MTHEN825

820 POKEK,32:K=K+D:IFPEEK(K)<>102AND
DTHENPOKEK,99:FORT=1TO99:NEXT:GOTO82
0
825 IFD=-1THEN840
830 PRINT"[HOM][3CRSRD]"SPC(M);"[SPC]
[SHIFTM]O[SPC][4CRSRL][CRSRD][SPC][COMM@]
[COMM+][SHIFTM][4CRSRL][CRSRD][2COMM@]
[SHIFTN][COMM@]":FORT=1TO950:NEXT
835 PRINT"[HOM][3CRSRD]"SPC(M);"[4SPC]
[4CRSRL][CRSRD][2SPC][COMM@][SPC][4CRSRL]
[CRSRD][COMM@][SHIFTN][COMM+]O[4CRSRL]
[CRSRD][SPC][3COMMT]":FORT=1TO500:NE
XT:GOTO850
840 PRINT"[HOM][3CRSRD]"SPC(M);"[SPC]
O[SHIFTN][SPC][4CRSRL][CRSRD][SHIFTN]
[COMM+][COMM@][SPC][4CRSRL][CRSRD][COMM@]
[SHIFTM][2COMM@]":FORT=1TO950:NEXT
845 PRINT"[HOM][3CRSRD]"SPC(M);"[4SPC]
[4CRSRL][CRSRD][SPC][COMM@][2SPC][4CRSRL]
[CRSRD]O[COMM+][SHIFTM][COMM@][4CRSRL]
[CRSRD][3COMMT][SPC]":FORT=1TO500:NE
XT:GOTO850
850 FORT=0TO4000:NEXT:RUN
```



Botón RESET

¿No sabes cómo hacerte un botón RESET? Puedes hacerte uno desde el teclado escribiendo:

Poke 792, peek(65532):poke 793,peek(65533)

Cada vez que pulsemos la tecla RESTORE, el ordenador hará reset. Cuidado porque los programas que tengamos almacenados se borrarán.

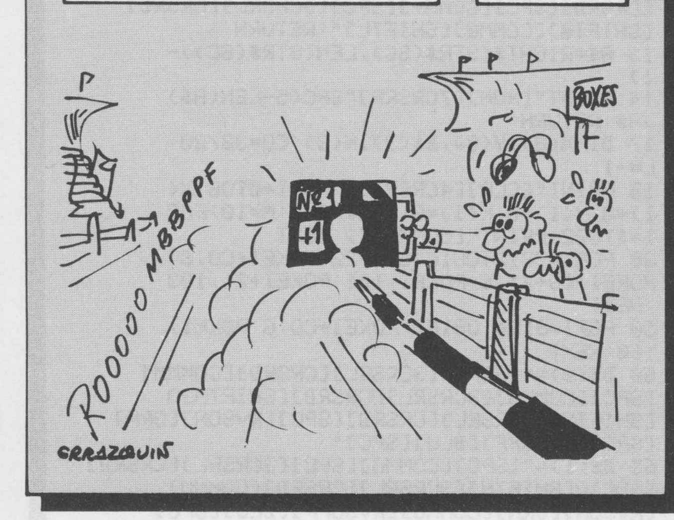
Funciona al igual en el VIC y en el 64.

B.G.R. RUN EE.UU.

Fórmula 1

Andrés Sánchez
Apdo. 6.064. C/Postal 50.007
Zaragoza

C-64



Es necesario un joystick en el port nº 2 (junto al interruptor. El coche se puede mover hacia arriba y abajo, izquierda y derecha.
Si chocamos contra los lados u otro coche se oye una explosión y comenzar.
Podéis cambiar los colores y el tiempo de juego (líneas 21, 22 y 54). P es la puntuación.

```
2 REM **** 1984 ****
3 PRINT"[CLR]"
4 PRINTTAB(15)"[2CRSRD]FORMULA[SPC]-
[SPC]1"
6 PRINT"[2CRSRD]NUESTRO[SPC]COCHE[SPC]
SE[SPC]DESPLAZA[SPC]CON[SPC]JOYSTICK
"
7 PRINTTAB(7)"[2CRSRD]←[SPC]IZQUIERD
A[SPC]*[SPC]DERECHA[SPC]-[SHIFTZ]"
8 PRINTTAB(10)"[2CRSRD]↑[SPC]ARRIBA[SPC]
*[SPC]ABAJO[SPC]!"
10 V=53248:POKEV+21,7
15 FORS1=12288TO12350:READQ1:POKES1,
Q1:NEXT
20 POKE53276,PEEK(53276)OR7
21 POKEV+39,1:POKEV+40,6:POKEV+41,7
22 POKEV+37,4:POKEV+38,0
23 REM *** COPIA DE RUTINA ***
24 FORX=49152TO49383:READA$:GOSUB500
:POKEX,D:NEXT
30 POKE53280,13:POKE53281,13
32 PRINT"[CLR]":P=0
34 POKE2040,192:POKE2041,192:POKE204
2,192
36 SYS 49152
40 FORX=1TO25
42 PRINT"[10COMM+][21SPC][8COMM+]"
44 NEXTX
50 TI$="000000"
52 SYS 49240
54 IFTI>10000THEN250
56 P=P+100
60 GOTO52
250 PRINT"[CLR]":POKEV+21,0
```

```
252 PRINTTAB(20)"[2CRSRD]TIEMPO[2CRSRD]
"
254 PRINTTAB(20)TI
256 PRINTTAB(20)"[2CRSRD]PUNTOS[2CRSRD]
"
258 PRINTTAB(20)P
260 PRINTTAB(5)"[2CRSRD]QUIERES[SPC]
JUGAR[SPC]DE[SPC]NUEVO[SPC]S/N"
262 GETA$:IFA$=""THEN262
264 IFA$="S"THENPOKEV+21,7:GOTO32
265 IFA$<>"N"THENPRINT"[2CRSRU]":GOT
O262
266 PRINT"[CLR]":END
298 REM ** DATAS DE SPRITE **
300 DATA0,40,0,240,40,15,240,40,15
302 DATA242,170,143,246,150,159,240,
150,15
304 DATA240,170,15,0,105,0,0,105,0
306 DATA0,105,0,0,105,0,0,105,0
308 DATA0,105,0,0,105,0,240,85,15
310 DATA240,190,15,246,190,159,242,1
70,143
312 DATA240,40,15,240,40,15,0,40,15
350 REM ******** RUTINA C/M ********

351 REM ** SONIDO CHOQUE DE SPRITES
**
352 DATAA2,10,A0,00,84,FB,CA,8E,10,D
4,A9,81
354 DATA8D,04,D4,A9,0F,8D,05,D4,A9,2
4,8D,01,D4
356 DATAA9,50,8D,00,D4,A0,00,C8,C0,F
F,D0,FB
358 DATAA4,FB,C8,84,FB,C0,40,D0,D9,E
0,00
360 DATAD0,D0,A9,00,8D,04,D4,8D,05,D
4
361 REM ** POSICION INICIAL DE SPRIT
ES *
362 DATAA2,D2,8E,00,D0,A0,C8,8C,01,D
0,A2,9B
364 DATA8E,02,D0,A0,28,8C,03,D0,A2,F
1,8E,04,D0
366 DATAA0,91,8C,05,D0
367 REM ** CHOQUE DE SPRITE 1 **
368 DATAAD,1E,D0,29,01,C9,01,F0,9F,A
D,1F,D0
370 DATA29,01,C9,01,F0,96
371 REM * CONTROL DE JOYSTICK Y SPRI
TE 1 *
372 DATAAE,00,D0,AD,00,DC,C9,7B,D0,0
3,CA,CA,CA
374 DATAC9,77,D0,03,E8,E8,E8,8E,00,D
0,AC,01,D0,C9
376 DATA7D,D0,02,C8,C8,C9,7E,D0,02,8
8,88,8C,01,D0
377 REM * MOVIMIENTO SPRITES 2 Y 3 *

378 DATAAE,02,D0,8E,02,D0,AC,03,D0
380 DATAC8,C8,C8,C8,C8,8C,03,D0,C0,F
0,F0,16
382 DATAAE,04,D0,8E,04,D0,AC,05,D0,C
8,C8,C8
384 DATAC8,C8,8C,05,D0,C0,F0,F0,16,6
0
386 DATAA0,0A,8C,03,D0,E0,9B,D0,06,A
2,C6
388 DATA8E,02,D0,60,A2,9B,8E,02,D0,6
0
390 DATAA0,28,8C,05,D0,E0,F1,D0,06,A
2,70
392 DATA8E,04,D0,60,A2,F1,8E,04,D0,6
0
500 D=0:FORJ=1TO2:D%=ASC(A$):D%=D%-4
8+(D%>64)*7:A$=MID$(A$,2):D=16*D+D%:
NEXT
502 RETURN
3000 ,40,0,240,40,15,240,40,15
```

Rescate lunar

Alvaro Ibáñez
Cardenal Siliceo, 20
28002 Madrid

VIC-20 sin expansión

El objetivo de este juego es rescatar a 6 astronautas que se encuentran atrapados en la Luna. Para ello hemos de bajar con nuestro módulo desde la nave nodriza y aterrizar en una de las tres plataformas que hay en el suelo.
Entonces hemos de volver a la nave nodriza, pero hemos de tener cuidado porque unos ovnis que recorren la pantalla intentarán impedírnoslo disparando rayos destructores.
Nuestra nave se controla con: Z=izquierda, C=derecha y M para disparar a los ovnis o para activar los propulsores mientras bajamos, para evitar los asteroides.
Cuando tecleéis el programa debéis tener en cuenta que algunas líneas ocupan más de los 88 caracteres permitidos, por lo que tendréis que utilizar algunas de las abreviaturas de las palabras clave.

```
1 DIMP(5),J(5),S(2),K(2),R$(6),L$(6)
:POKE36879,110:POKE650,255
2 PRINT"[CLR][BLU][RVSON][4SPC]RESCA
TE[SPC]LUNAR[5SPC]":FORI=7680TO7701:
POKEI+30720,1:FORT=1TO90:NEXT:NEXT:G
OTO32
7 REM(C)A.IBANEZ
8 PRINT"[HOM][17CRSRD][YEL]T[WHT][20SPC]
[YEL]T[WHT][RVSON][RED][SPC][RVSOFF]
[YEL]T[WHT][18SPC][YEL]T[WHT][RVSON]
[RED][SPC][RVSOFF]";
10 PRINT     "[RVSON][RED][2SPC][RVSOFF]
[YEL]T[WHT][7SPC][RVSON]100[RVSOFF][6SPC]
[YEL]T[RVSON][RED][2SPC][RVSOFF]";
11 PRINT     "[RVSON][3SPC][RVSOFF][2SPC]
[RVSON][WHT]150[RVSOFF][2SPC][RVSON]
150[RVSOFF][6SPC][RVSON][RED][3SPC][RVSOFF]
[RVSON][RED][3SPC][RVSOFF][2SPC][RVSON]
[WHT]250[RVSOFF][2SPC][RVSON]250[RVSOFF]
[2SPC][RVSON]250[RVSOFF][SPC][RVSON]
[RED][3SPC][RVSOFF]";
12 PRINT     "[RVSON][RED][21SPC][RVSOFF]
";:GOTO60
32 N$="[YEL][SPC][COMML][COMMY][COMMJ]
[SPC][5CRSRL][CRSRD][SPC][RVSON][COMML]
[RVSOFF][WHT][SPC][YEL][RVSON][COMMJ]
[RVSOFF][SPC][5CRSRL][CRSRD][SPC][3COMMT]
```

```
[SPC]":S$="[HOM][YEL][SPC]SCORE[WHT]
":V$="[HOM][PUR][14CRSRR]"
38 H$="[HOM][WHT][CRSRD][6CRSRR]":F$
="[HOM][YEL][CRSRD][14CRSRR]FUEL[WHT]
":L$="[HOM][5CRSRD]":GOTO2000
49 F$="[HOM][YEL][CRSRD][14CRSRR]FUE
L[WHT]":NV=3:SC=0:S(0)=1:S(1)=-1:S(2
)=0:TT=200:SP=50
50 N=0:FU=800:NH=0:TT=TT-50:SP=SP+50
:PRINT"[CLR][7CRSRD][WHT][6SPC]T[SPC]
="SP:FORT=1TO2000:NEXT:GOTO8
60 W=0:M=7749:K(1)=INT(RND(1)*9+1):X
=1:K(2)=INT(RND(1)*9+110):BS=22:F=1:
FORI=1TO100:NEXT
65 PRINT"[HOM][CRSRD]";:FORI=1TO16:P
RINT"[22SPC]";:NEXT:FORI=0TO5
67 J(I)=0:P(I)=7815+INT(RND(1)*200):
NEXT:E=1:N=N+1:IFN>6THENSC=SC+(SP*NH
):GOTO50
70 PRINTS$;SC:PRINTF$;FU:IFNH>0THENP
RINTH$;:FORI=1TONH:PRINT"[YEL]T";:NE
XT
72 POKE38905,2:POKE8185,160:PRINTV$;
"[SPC][CRSRL]";:IFNV>0THENFORI=1TONV
:PRINT"[SHIFTQ]";:NEXT:PRINT"[SPC]"
80 PRINT"[HOM][2CRSRD]";SPC(E);N$:PO
KEM,1:GETA$:IFA$="M"ORW>52ANDRND(1)>
.94THEN90
86 E=E+X:M=M+X:W=W+1:IFE>17ORE<1THEN
X=X*-1
87 FORT=1TOTT:NEXT:GOTO500
90 FORI=M+21TOM+23:POKEI,32:FORT=1TO
100:NEXT:NEXT:GOTO100
95 POKEM,1:POKEM-BS-D,32:IFPEEK(M-1)
=246THENSC=SC+SP:NH=NH+1:GOTO60
96 FORI=M+44TO8076STEP22:POKEI-22,32
:POKEI+30720,1:POKEI,20:FORT=1TO99:N
EXT:NEXT:POKEI-22,32:GOTO60
100 GETA$:D=(A$="Z")-(A$="C"):IFA$="
M"ANDFU>0THEN200
140 M=M+BS+D:IFM<7768THEN95
145 IFPEEK(M)>160THEN900
150 IFPEEK(M)<>32ANDPEEK(M)<>1THEN80
0
160 POKEM+30720,7:POKEM,1:POKEM-BS-D
,32:GOTO500
200 IFF=2THENI=M-22:GOTO210
205 E=E+1:IFE>2THENE=0:FU=FU-INT(RND
(1)*80+50):PRINTF$;FU;"[CRSRL][SPC][WHT]
";:D=0:GOTO140
207 GOTO145
210 I=I-22:IFPEEK(I)=60ORPEEK(I)=62T
HEN220
215 POKEI,46:POKEI+22,32:IFI<7813THE
NPOKEI,32:GOTO140
217 GOTO210
220 FORE=0TO5:IFP(E)=ITHENP(E)=0:POK
EI,160:POKEI,42:POKEI,32:POKEI+22,32
:SC=SC+100:PRINT
230 IFJ(E)<>0THENPOKEJ(E),32
240 NEXT
500 IFF=2THEN510
502 K(1)=K(1)+1:PRINTL$;SPC(K(1))L$(
N):K(2)=K(2)-1:PRINTL$;SPC(K(2))R$(N
)
503 IFM<7768THEN80
504 FORT=1TOTT:NEXT:IFPEEK(M)<>1ANDP
EEK(M)<>32THEN800
506 GOTO100
510 FORI=0TO5:IFP(I)=0THEN540
515 P(I)=P(I)+S(INT(I/3)):POKEP(I),6
1+S(INT(I/3)):POKEP(I)-S(INT(I/3)),3
2:IFJ(I)<>0THEN530
525 IFRND(1)>.9THENJ(I)=P(I)+22:POKE
J(I),66:GOTO540
530 IFJ(I)<>0THENJ(I)=J(I)+22:POKEJ(
I),66:POKEJ(I)-22,32:IFJ(I)>8053THEN
POKEJ(I),32:J(I)=0
540 IFPEEK(M)=1ORPEEK(M)=32THENNEXT:
GOTO100
```

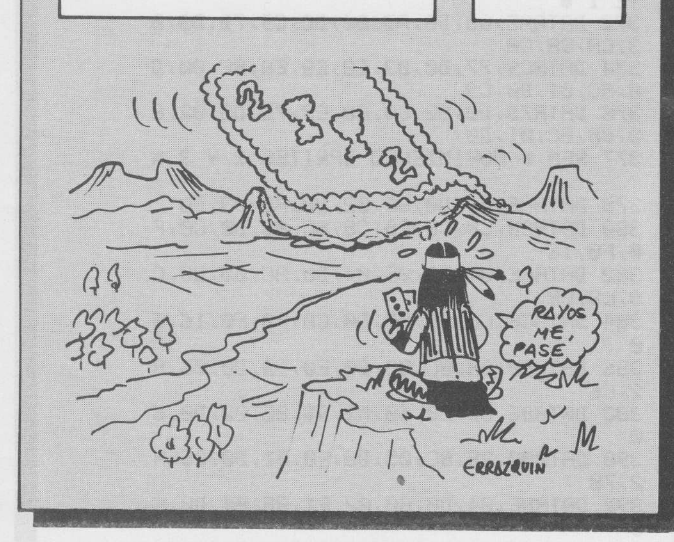
```
800 POKEM-BS-D,32:FORI=1TO10:POKEM,1
60:FORT=1TO9:NEXT:POKEM,1:FORT=1TO9:
NEXT:NEXT:NV=NV-1:IFF=1THENN=N-1
820 IFNV>=0THEN60
821 PRINTS$;SC:FORT=1TO5000:NEXT:RUN

900 SC=SC+100:IFM>8120THENSC=SC+50:I
FM>8142THENSC=SC+100
910 POKEM-22-D,32:PRINTS$;SC:POKEM-2
,32:POKEM-1,32:POKEM,1:POKEM+1,32:PO
KEM+2,32
930 PRINT"[HOM][4CRSRD]";:FORI=1TO13
:PRINT"[22SPC]";:NEXT:FORT=1TO1000:N
EXT:PRINTS$;SC
950 BS=-22:F=2:GOTO100
2000 L$(1)="[SPC][SHIFTW][2CRSRD][2CRSRR]
[SPC][SHIFTW][4CRSRR][3CRSRD][SPC][SHIFTW]
[3CRSRR][2CRSRU][SPC][SHIFTW]":R$(1)
="[CRSRR][3CRSRD][SHIFTW][SPC][2CRSRR]
[3CRSRD][SHIFTW][SPC][3CRSRR][2CRSRU]
[SHIFTW][SPC][3CRSRR][CRSRU][SHIFTW]
[SPC]"
2004 L$(2)=L$(1)+"[6CRSRR][SPC][SHIFTW]
":L$(3)=L$(2)+"[3CRSRR][SPC][SHIFTW]
":L$(4)=L$(3)+"[CRSRR][3CRSRU][SPC][SHIFTW]
"
2010 L$(5)=L$(4)+"[CRSRR][CRSRD][SPC]
[SHIFTW]":L$(6)=L$(5)+"[3CRSRR][2CRSRD]
[SPC][SHIFTW]":R$(2)=R$(1)+"[2CRSRR]
[2CRSRD][SHIFTW][SPC][CRSRL]"
2016 R$(3)=R$(2)+"[2CRSRR][2CRSRU][SHIFTW]
[SPC][CRSRL]":R$(4)=R$(3)+"[2CRSRR][CRSRD]
[SHIFTW][SPC][CRSRL]":R$(5)=R$(4)+"[3CRSRR]
[4CRSRU][SHIFTW][SPC][CRSRL]"
2022 R$(6)=R$(5)+"[6CRSRR][5CRSRD][SHIFTW]
[SPC][CRSRL]":GOTO49
```

Siete y media

Rafael Alberto Garcia Medina
Carretera de Granada, nº 4, 3º C
Guadix (Granada)

C-64



Este programa es cortísimo pero muy divertido. Todo consiste en tener un poco de suerte y saber plantarse a su debido tiempo. Tienes un 30% de que te salga la media y un 10% en los demás núm. El programa está basado en el popular juego de cartas llamado de igual nombre.
El ordenador hace de banca y te da cartas hasta que te plantes.

Más tarde él descubre su carta y entonces pueden pasar dos casos:

—Si la puntuación de su carta es mayor o igual a la tuya gana la banca (el ordenador).

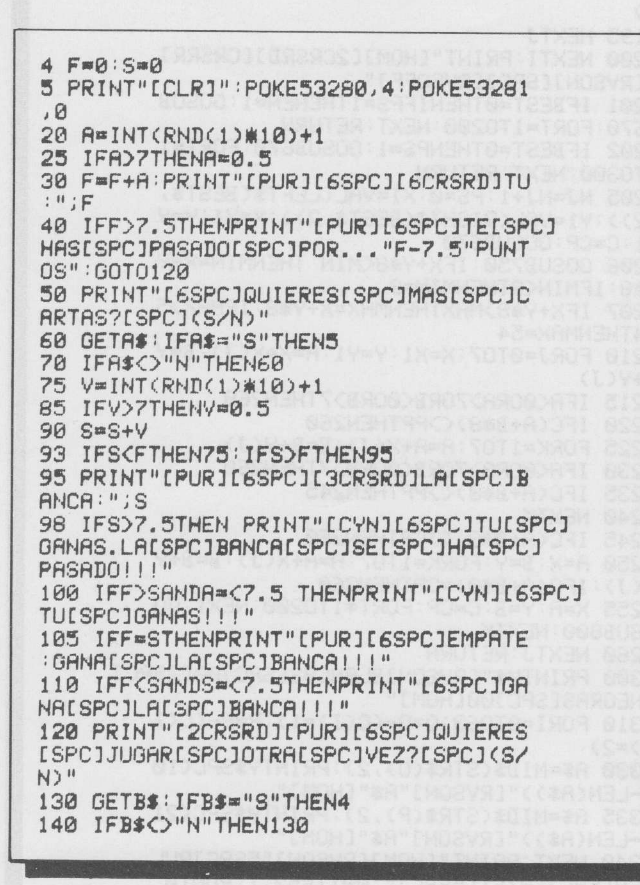
—Si la puntuación de su carta es menor que la tuya entonces ganas tú.

Pero ojo, ya que si te pasas de 7,5 pierdes.

Es muy interesante el programa ya que lo he programado (válgame la redundancia) con el objetivo de que el ordenador no se dé por aludido hasta que la puntuación de sus cartas sea mayor o igual a la del jugador, pero claro: hay veces en las que también se pasa por avaricioso.

Cuidado con el refrán: LA AVARICIA ROMPE EL SACO.

¡¡¡Suerte y a jugar contra la máquina!!!



```
4 F=0:S=0
5 PRINT"[CLR]":POKE53280,4:POKE53281
,0
20 A=INT(RND(1)*10)+1
25 IFA>7THENA=0.5
30 F=F+A:PRINT"[PUR][6SPC][6CRSRD]TU
:";F
40 IFF>7.5THENPRINT"[PUR][6SPC]TE[SPC]
HAS[SPC]PASADO[SPC]POR..."F-7.5"PUNT
OS":GOTO120
50 PRINT"[6SPC]QUIERES[SPC]MAS[SPC]C
ARTAS?[SPC](S/N)"
60 GETA$:IFA$="S"THEN5
70 IFA$<>"N"THEN60
75 V=INT(RND(1)*10)+1
85 IFV>7THENV=0.5
90 S=S+V
93 IFS<FTHEN75:IFS>FTHEN95
95 PRINT"[PUR][6SPC][3CRSRD]LA[SPC]B
ANCA:";S
98 IFS>7.5THEN PRINT"[CYN][6SPC]TU[SPC]
GANAS.LA[SPC]BANCA[SPC]SE[SPC]HA[SPC]
PASADO!!!"
100 IFF>SANDA=<7.5 THENPRINT"[CYN][6SPC]
TU[SPC]GANAS!!!"
105 IFF=STHENPRINT"[PUR][6SPC]EMPATE
:GANA[SPC]LA[SPC]BANCA!!!"
110 IFF<SANDS=<7.5THENPRINT"[6SPC]GA
NA[SPC]LA[SPC]BANCA!!!"
120 PRINT"[2CRSRD][PUR][6SPC]QUIERES
[SPC]JUGAR[SPC]OTRA[SPC]VEZ?[SPC](S/
N)"
130 GETB$:IFB$="S"THEN4
140 IFB$<>"N"THEN130
```



Detección de errores en disco

Cuando el LED de la unidad de discos se encuentra destellando es señal de que algún error se ha dado mientras estabas usando el disco. Para averiguar cuál es, añade la siguiente línea a tu programa:

0 OPEN15,8,15:INPUT@15,A,B$:PRINT A,B$:CLOSE15:END

Ahora haz RUN. El LED dejará de parpadear y en la pantalla aparecerá el tipo de error que has cometido y una breve descripción.

En las páginas 69-74 del manual del usuario se encuentra esta lista de errores.

R.A.A. RUN EE.UU.



Othello

Alvaro Ibáñez
Cardenal Siliceo, 20
28002 Madrid

VIC-20 +3K superexpander

El Othello es un juego de inteligencia, del estilo del "Go" o de las damas.

Cómo se juega al Othello

Para los que no sepáis cómo se juega, explicaré las reglas.

El tablero consta de 8×8=64 casillas, todas del mismo color.

Hay dos jugadores, y cada uno tiene las fichas de un color; blancas o negras. Comienzan siempre las negras.

Al principio del juego hay colocadas cuatro fichas (dos de cada color) en el centro del tablero.

Se juega por turnos, colocando una ficha del color propio. Cada ficha que se coloca ha de capturar al menos a una ficha contraria.

La captura se realiza cuando dos fichas propias encierran a una o más fichas contrarias en cualquier dirección, también en diagonal. Si no se puede capturar ninguna ficha, se cede el turno al contrario.

Las fichas que se capturan no se retiran del tablero, sino que cambian de color, de modo que las fichas contrarias que se capturan pasan a ser fichas propias.

Se sigue jugando hasta que se ocupan todas las casillas del tablero o hasta que ninguno de los jugadores pueda hacer un movimiento legal.

El ganador es el que más fichas tiene cuando se acaba la partida.

El programa

El programa ofrece varias opciones:

1.—Un jugador: Jugamos nosotros contra el ordenador.

2.—Dos jugadores: Para que jueguen dos personas una contra otra.

3.—Programación: Para programar posiciones y comenzar a jugar a partir de ahí.

4.—Demostración: El ordenador juega contra sí mismo para enseñarnos cómo se hace.

Si elegimos la opción 1 (la más interesante) podremos jugar contra el ordenador. El ordenador no juega simplemente a comer la mayor cantidad de fichas posibles, sino que tiene en cuenta la altura del juego a la que estamos, y hace una valoración del tablero dependiendo de cómo vaya la partida. Se puede decir que a un jugador medio le gana la mitad de las veces.

En las otras opciones el ordenador nos hará preguntas como ¿tu color?, ¿quién juega primero?, ¿dónde colocas la ficha?, etc.

Para decirle dónde vamos a mover, basta con indicarle la fila y la columna (a-h y 1-8). No hace falta que pulsemos RETURN. Cuando no podemos mover, tenemos que decirle que movemos a Ø, Ø.

```
7 READA$:IFA$<>"E"THEN7
8 POKE56,28:POKE52,28:POKE51,0:POKE5
5,0
9 READC:IFCTHENFORJ=0TO7:READK:POKE7
168+C*8+J,K:NEXT:GOTO9
10 DIMC(63),M(63),X(7),Y(7),A$(2):PR
INT"[CLR]":POKE36869,240:RESTORE
12 FORI=0TO7:READX(I),Y(I):NEXT:GOSU
B650:A$(0)="@A[2CRSRL][CRSRD]BC":A$(
1)="DE[2CRSRL][CRSRD]FG":A$(2)="HI[2CRSRL]
[CRSRD]JK
23 V=0:EN=0:MIN=27:MAX=36:TU=1
24 Y$="[HOM][22CRSRD]":S$="[RVSON][6SPC]
[SPC][5SPC][SPC][8SPC][HOM]":GOTO400

25 POKE36879,25:POKE36869,255:PRINT"
[CLR][BLU][CRSRR]";
30 FORI=1TO9:PRINTA$(2)"[CRSRU]";:NE
XT:PRINT:PRINT
40 FORI=1TO8:PRINT"[CRSRR]"A$(2)"[CRSRU]
";:FORJ=1TO8:PRINTA$(2)"[CRSRU]";:NE
XT
45 PRINT"[CRSRD][RVSON][CRSRR]"MID$(
STR$(9-I),2):NEXT
50 PRINT"[CRSRD][4CRSRR][RVSON]A[CRSRR]
B[CRSRR]C[CRSRR]D[CRSRR]E[CRSRR]F[CRSRR]
G[CRSRR]H
51 IFS=3THENRETURN
55 C=1:X=3:Y=3:GOSUB800:X=4:Y=4:GOSU
B800:C=2:X=4:Y=3:GOSUB800:X=3:Y=4:GO
SUB800
57 RETURN
60 IFNJ=60THENEN=1:RETURN
61 GOSUB660:SW=0:GOSUB700
65 IFT$="00"THENIFPS=1THENEN=1:GOSUB
670:FORT=1TO200:NEXT:RETURN
66 IFT$="00"THENPS=1:GOSUB670:FORT=1
TO200:NEXT:RETURN
70 PS=0:NJ=NJ+1:X1=ASC(LEFT$(T$,1))-
65:Y1=8-VAL(RIGHT$(T$,1)):C=PP:X=X1:
Y=Y1
71 IFC(X+Y*8)=0THEN73
72 PRINTY$"[RVSON][3SPC]MOVIMIENTO[SPC]
ILEGAL[HOM]":FORT=1TO1500:NEXT:PRINT
Y$S$:NJ=NJ-1:TU=3-TU:GOTO60
73 GOSUB670:GOSUB750:IFX+Y*8<MIN THE
NMIN=X+Y*8:IFMIN<9THENMIN=9
74 IFX+Y*8>MAXTHENMAX=X+Y*8:IFMAX>54
THENMAX=54
75 FORJ=0TO7:X=X1:Y=Y1:A=X+X(J):B=Y+
Y(J)
80 IFA<0ORA>7ORB<0ORB>7THEN97
85 IFC(A+B*8)<>CPTHEN97
90 FORK=1TO7:A=A+X(J):B=B+Y(J)
91 IFA<0ORA>7ORB<0ORB>7THEN97
92 IFC(A+B*8)<>CPTHEN94
93 NEXTK
94 IFC(A+B*8)<>PPTHEN97
95 A=X:B=Y:FORK=1TO7:A=A+X(J):B=B+Y(
J):IFC(A+B*8)=PPTHEN97
96 X=A:Y=B:C=PP:FORT=1TO200:NEXT:SW=
1:GOSUB800:NEXTK
97 NEXTJ:IFSW=1THENRETURN
98 PRINTY$"[RVSON][3SPC]MOVIMIENTO[SPC]
ILEGAL[HOM]":FORT=1TO1500:NEXT:PRINT
Y$S$:NJ=NJ-1:TU=3-TU
99 C=3:X=X1:Y=Y1:GOSUB800:GOTO60
100 IFNJ=60THENEN=1:RETURN
101 GOSUB660:PRINT"[HOM][2CRSRD][CRSRR]
[SPC]"
102 IFNJ>20ANDJ2=0THENGOSUB650:J2=1
103 IFNJ>40ANDJ3=0THENGOSUB650:J3=1:
V=1
110 BEST=0:FORI=MIN-9TOMAX+9
115 IFC(I)<>0THEN200
120 Y=INT(I/8)
125 X=I-Y*8
130 TP=0:FORJ=0TO7:A=X:B=Y
135 A=A+X(J):B=B+Y(J):IFA<0ORA>7ORB<
0ORB>7THEN195
140 IFC(A+B*8)<>PPTHEN195
145 FORK=1TO7:A=A+X(J):B=B+Y(J):IFA<
0ORA>7ORB<0ORB>7THEN195
150 IFC(A+B*8)<>PPTHEN160
155 NEXTK
160 IFC(A+B*8)<>CPTHEN195
165 TP=TP+M(I)-K*(V=1)-(1/K)*(V=0)
170 IF(TP>BEST)OR((TP=BEST)ANDRND(1)
>.5)THENBEST=TP:BEST$=STR$(X)+STR$(Y
)
195 NEXTJ
200 NEXTI:PRINT"[HOM][2CRSRD][CRSRR]
[RVSON][SPC][RVSOFF]"
201 IFBEST=0THENIFPS=1THENEN=1:GOSUB
670:FORT=1TO200:NEXT:RETURN
202 IFBEST=0THENPS=1:GOSUB670:FORT=1
TO300:NEXT:RETURN
205 NJ=NJ+1:PS=0:X1=VAL(LEFT$(BEST$,
2)):Y1=VAL(RIGHT$(BEST$,2)):X=X1:Y=Y
1:C=CP:GOSUB670
206 GOSUB750:IFX+Y*8<MIN THENMIN=X+Y
*8:IFMIN<9THENMIN=9
207 IFX+Y*8>MAXTHENMAX=X+Y*8:IFMAX>5
4THENMAX=54
210 FORJ=0TO7:X=X1:Y=Y1:A=X+X(J):B=Y
+Y(J)
215 IFA<0ORA>7ORB<0ORB>7THEN260
220 IFC(A+B*8)<>PPTHEN260
225 FORK=1TO7:A=A+X(J):B=B+Y(J)
230 IFA<0ORA>7ORB<0ORB>7THEN260
235 IFC(A+B*8)<>PPTHEN245
240 NEXTK
245 IFC(A+B*8)<>CPTHEN260
250 A=X:B=Y:FORK=1TO7:A=A+X(J):B=B+Y
(J):IFC(A+B*8)=CPTHEN260
255 X=A:Y=B:C=CP:FORT=1TO200:NEXT:GO
SUB800:NEXTK
260 NEXTJ:RETURN
300 PRINTY$"[RVSON]BLANCAS[SPC]00[2SPC]
NEGRAS[SPC]00[HOM]"
310 FORI=0TO63:O=O-(C(I)=1):P=P-(C(I
)=2)
330 A$=MID$(STR$(O),2):PRINTY$SPC(10
-LEN(A$))"[RVSON]"A$"[HOM]"
335 A$=MID$(STR$(P),2):PRINTY$SPC(21
-LEN(A$))"[RVSON]"A$"[HOM]"
340 NEXT:PRINT"[HOM][RVSON][5SPC]PUL
SA[SPC]SHIFT[5SPC]":WAIT653,1:RUN10
400 POKE36879,8:PRINT"[HOM][RVSON][BLK]
[8SPC]OTHELLO[7SPC][WHT]
405 FORI=38400TO38421:POKEI,1:FORT=1
TO90:NEXT:NEXT
410 PRINT"[CRSRD][SPC](C)ALVARO[SPC]
IBANEZ
415 PRINT"[2CRSRD][5SPC]ELIJE [RVSOFF]

420 PRINT"[2CRSRD][2SPC][RVSON]1[RVSOFF]
[SPC]UN[SPC]JUGADOR
425 PRINT"[2CRSRD][2SPC][RVSON]2[RVSOFF]
[SPC]DOS[SPC]JUGADORES
430 PRINT"[2CRSRD][2SPC][RVSON]3[RVSOFF]
[SPC]PROGRAMAR
435 PRINT"[2CRSRD][2SPC][RVSON]4[RVSOFF]
[SPC]DEMOSTRACION
440 GETA$:IFVAL(A$)<1ORVAL(A$)>4THEN
440
441 S=VAL(A$):FORI=1TO5:POKE7814+S*6
6,32:FORT=1TO150:NEXT:POKE7814+S*66,
176+S
442 FORT=1TO200:NEXT:NEXT:GOSUB25
443 ONSGOTO450,460,470,500
450 PRINTY$"[RVSON]TU[SPC]COLOR[SPC]
(B/N)?[SPC][RVSOFF][SPC][CRSRL]";:CH
=0:H=15
```

```
451 GOSUB690:IFA$<>"B"ANDA$<>"N"THEN
H=15:CH=0:GOTO451
452 PRINT"[RVSON]"A$"[RVSOFF][HOM]":
CP=2:PP=1:IFA$="N"THENPP=2:CP=1
453 FORT=1TO999:NEXT:PRINTY$S$
455 IFCP=2THEN457
456 GOSUB60:IFEN=1THEN300
457 GOSUB100:IFEN=1THEN300
458 GOTO456
460 PP=2:CP=1:GOSUB60:IFEN=1THEN300
465 PP=1:CP=2:GOSUB60:IFEN=1THEN300
469 GOTO460
470 PRINTY$S$Y$"[RVSON]LUGAR?[SPC][RVSOFF]
[SPC][RVSON][CRSRL]";:GOSUB701:IFT$=
"00"THEN490
475 NJ=NJ+1:PRINTY$"[RVSON][11CRSRR]
COLOR?[SPC][RVSOFF][SPC][CRSRL]";:GO
SUB690
480 X=ASC(LEFT$(T$,1))-65:Y=8-VAL(RI
GHT$(T$,1)):C=1:IFA$="N"THENC=2
485 GOSUB800:PRINTY$S$:FORT=1TO300:N
EXT:GOTO470
490 PRINTY$S$Y$"[RVSON]TU[SPC]COLOR(
B/N)?[SPC][RVSOFF][SPC][CRSRL]";:GOS
UB690:PP=1::CP=2:IFA$="N"THENPP=2:CP
=1
495 PRINTY$S$Y$"[RVSON]QUIEN[SPC]MUE
VE[SPC](B/N)?[SPC][RVSOFF][SPC][CRSRL]
";:GOSUB690:PRINTY$S$
496 MIN=9:MAX=54:TU=1:IFA$="B"THENTU
=2
497 IF(A$="B"ANDPP=1)OR(A$="N"ANDPP=
2)THEN456
498 GOTO457
500 CP=2:PP=1:GOSUB100:IFEN=1THEN300
505 CP=1:PP=2:GOSUB100:IFEN=1THEN300
510 GOTO500
650 FORI=0TO7:READA$:FORJ=0TO7:M(I*8
+J)=ASC(MID$(A$,J+1,1))-64:NEXT:NEXT
:RETURN
660 TU=3-TU:IFTU=1THENPRINT"[HOM][4CRSRR]
[RVSON]MUEVE[SPC]EL[SPC]BLANCO[RVSOFF]
":RETURN
665 PRINT"[HOM][4CRSRR][RVSON]MUEVE[SPC]
EL[SPC]NEGRO[RVSOFF]":RETURN
670 PRINT"[HOM]"S$:RETURN
690 GETA$:IFA$="B"ORA$="N"THENRETURN
691 H=H+1:IFH<15THEN690
692 H=0:IFCH=1THENPRINT"[RVSON][SPC]
[RVSOFF][CRSRL]";:CH=0:GOTO690
693 CH=1:PRINT"[RVSOFF][SPC][CRSRL]"
;:GOTO690
700 PRINTY$S$Y$"[RVSON][SPC]MOVIMIEN
TO?[SPC][RVSOFF][SPC][CRSRL]";
701 POKE198,0:CH=1:T$=""
705 GETA$:IFA$=""THEN740
707 IFASC(A$)=20ANDLEN(A$)=1THENT$="
":PRINT"[RVSOFF][CRSRL][SPC][RVSON][SPC]
[2CRSRL]";:CH=0:H=0:GOTO705
710 IFT$=""THENIF(A$<"A"ORA$>"H")AND
A$<>"0"THEN705
715 IFLEN(T$)=1THENIF(VAL(A$)>8ORVAL
(A$)=0)ANDA$<>"0"THEN705
720 IFLEN(T$)=1THENIF(A$="0"ANDT$<>"
0")OR(A$<>"0"ANDT$="0")THEN705
725 PRINT"[RVSON]"A$"[RVSOFF]";:T$=T
$+A$:IFLEN(T$)=2THENRETURN
730 CH=0:H=15
740 H=H+1:IFH<15THEN705
741 H=0:IFCH=0THENCH=1:PRINT"[RVSON]
[SPC][RVSOFF][CRSRL]";:GOTO705
742 CH=0:PRINT"[SPC][CRSRL]";:GOTO70
5
750 S=C:FORJ=1TO4:C=S:GOSUB800:FORT=
1TO200:NEXT:C=3:GOSUB800:FORT=1TO150
:NEXT:NEXT
776 C=S:GOSUB800:RETURN
800 PRINTLEFT$(Y$,Y*2+3)SPC(X*2+3)A$
(C-1):C(X+Y*8)=C+C*(C=3):RETURN
849 DATA-1,-1,0,-1,1,-1,-1,0,1,0,-1,
1,0,1,1,1
850 DATA TAEDDEAT
851 DATA A@CBBC@A
852 DATA ECFDDFCE
853 DATA DBD@@DBD
854 DATA DBD@@DBD
855 DATA ECFDDFCE
856 DATA A@CBBC@A
857 DATA TAEDDEAT
860 DATA TAGFFGAT
861 DATA A@BEEB@A
862 DATA GBGCCGBG
863 DATA FEC@@CEF
864 DATA FEC@@CEF
865 DATA GBGCCGBG
866 DATA A@BEEB@A
867 DATA TAGFFGAT
870 DATA TACAACAT
871 DATA A@CBBC@A
872 DATA CCBAABCC
873 DATA ABA@@ABA
874 DATA ABA@@ABA
875 DATA CCBAABCC
876 DATA A@CBBC@A
877 DATA TACAACAT,E
900 DATA.1,0,3,12,16,32,32,64,64
901 DATA1,0,192,48,8,4,4,2,2
902 DATA2,64,64,32,32,16,12,3,0
903 DATA3,2,2,4,4,8,48,192,1
904 DATA4,0,3,15,31,63,63,127,127
905 DATA5,0,192,240,248,252,252,254
254
906 DATA6,127,127,63,63,31,15,3,0
907 DATA7,254,254,252,252,248,240,19
2,1
908 DATA8,0,0,0,0,0,0,0,0
909 DATA9,0,0,0,0,0,0,0,0
910 DATA10,0,0,0,0,0,0,0,0
911 DATA11,0,0,0,0,0,0,0,1
932 DATA32,255,255,255,255,255,255,2
55,255,0
```



Pokes simplificados

Esta idea es para el 64, pero se puede adaptar fácilmente al VIC-20.

En vez de hacer lo siguiente:

Poke 1024,81:poke55296,1 (para colocar un círculo en la esquina superior de la pantalla). Es más fácil y útil hacer:
co=54272

Poke 1024,81:poke1024+co,1

En el VIC, el 1024 es 7680 y la variable CO ha de ser 30720.

B.H.S. RUN EE.UU.

Hueco entre programas

Un buen truco es dejar 10 pasos de contador entre programa y programa cuando los grabes en una cinta. De este modo podrás ampliarlos y guardarlos en el mismo sitio sin tener que mover los demás programas.

A.L. RUN EE.UU.

Las Vegas

Rafael Alberto García Medina
Carretera de Granada, nº 4, 3º C
Guadix (Granada)

C-64

En este programa lo esencial es la buena suerte. Está basado en las máquinas de las frutas (máquinas que se encuentran en los bares y en los casinos), pero mucho más simple lógicamente.
El ordenador comienza haciéndote una pregunta a la cual le tienes que responder con el número al que pretendes llegar con las menores tiradas posibles.
Me explico:
Salen cuatro figuras moviéndose aleatoriamente en la pantalla y se paran pulsando una tecla. Si las figuras que has parado coinciden con las figuras de los premios, el ordenador te regala tantos puntos como se indique en el premio.
Se gana cuando se supera la puntuación fijada por el jugador en menos tiradas que las también fijadas por el mismo.
Se pierde cuando el número de tiradas sea mayor que las fijadas por el jugador al comienzo del juego.
Espero que todo esté claro. En la práctica el juego es fácil de aprender.
Que os divirtáis mucho y suerte, camaradas.

```
2 POKE53280,0:POKE53281,0
3 INPUT"[CLR][COMM6][5CRSRD]HASTA[SPC]
CUANTOS[SPC]PUNTOS[SPC]JUGAMOS";I
4 IFI<100THENPRINT"[RVSON][SPC]SON[SPC]
MUY[SPC]POCOS[SPC]PUNTOS":FORY=1TO15
00:NEXT:GOTO3
5 INPUT"Y[SPC]EN[SPC]CUANTAS[SPC]TIR
ADAS";O
7 PRINT"[CLR][PUR]PULSA[SPC]UNA[SPC]
TECLA[SPC]CUANDO[SPC]QUIERAS[SPC]PAR
AR":F=F+W
8 GOTO 300
9 E=E+1
10 PRINT"[BLU][RVSON][SPC]TIRADA[SPC]
";E:PRINT"[GRN]"
11 A=INT(RND(1)*4)+1
20 POKE55790+A,1
30 POKE1518+A,33
40 B=INT(RND(1)*4)+1
50 POKE55790+B,5
60 POKE1518+B,90
80 C=INT(RND(1)*4)+1
90 POKE55790+C,2
100 POKE1518+C,83
110 D=INT(RND(1)*4)+1
120 POKE55790+D,6
130 POKE1518+D,36
140 GETA$:IFA$=""THEN11
150 IFD=1ANDC=2ANDB=3ANDA=4THENPRINT
"1000[SPC]PUNTOS[SPC]!!!":W=1000:GOT
O170
160 IFA=1ANDB=2THENPRINT"500[SPC]PUN
TOS":W=500:GOTO170
163 IFB=1ANDC=2THENPRINT"500[SPC]PUN
TOS":W=500:GOTO170
164 IFD=4THENPRINT"20[SPC]PUNTOS":W=
20:GOTO170
165 IFC=1ORC=4THENPRINT"10[SPC]PUNTO
S":W=10:GOTO170
170 PRINT"[3CRSRD]JUEGAS[SPC]OTRA[SPC]
PARTIDA?[SPC](S/N)"
180 GETB$:IFB$="S"THEN7
190 IFB$<>"N"THEN180
199 END
300 PRINT"[2CRSRD][GRN][2SPC]*[SPC]P
REMIOS[SPC]*[6CRSRR][RVSON][BLU]PUNT
OS[SPC]TOTALES:";F:W=0
310 PRINT"[GRN][SHIFTS]---[SPC]=[SPC]
10[SPC]PUNTOS"
320 PRINT"---[SHIFTS][SPC]=[SPC]10[SPC]
PUNTOS"
325 PRINT"---$[SPC]=[SPC]20[SPC]PUNT
OS"
330 PRINT"![SHIFTZ]--[SPC]=500[SPC]P
UNTOS"
340 PRINT"[SHIFTZ][SHIFTS]--[SPC]=50
0[SPC]PUNTOS"
350 PRINT"$[SHIFTS][SHIFTZ]![SPC]=10
00[SPC]PUNTOS"
355 PRINT:PRINT
360 IFF=>ITHENPRINT"[RVSON][RED][SPC]
LO[SPC]HAS[SPC]CONSEGUIDO[SPC]EN[RVSOFF]
"E"[RVSON]TIRADAS[SPC]":FORY=1TO4000
:NEXT:RUN
365 IFE=OTHENPRINT"[RVSON][SPC]NO[SPC]
LO[SPC]HAS[SPC]CONSEGUIDO[SPC]EN"E"[RVSON]
TIRADAS[SPC]":FORY=1TO4000:NEXT:RUN
370 GOTO9
```



Basic Salomon

Cuando tres niños se están peleando por ver quién juega con el ordenador, es mejor que él mismo decida:

```
10 print "esperad mientras elijo"
20 fort=1to 2000:next
30 a=int(rn=(1)*3)+1
40 if a=1 then print"Jaime es el primero"
50 if a=2 then print"Pedro es el primero"
60 if a=3 then print"Ana es la primera"
```

N.L. RUN EE.UU.

Módulo

Para hallar el módulo de una función:

```
10 input"dividendo";a
20 input"divisor";b
30 def fn mod(d)=int((a/b)—int(a/b))*b+.5)
40 Print a "módulo "b"="fn mod (d)
```

P.L.V. RUN EE.UU.

Vic calc

Alvaro Ibáñez
Cardenal Siliceo, 20
28002 Madrid

VIC-20
+3K
superexpander
C-64



Este programa simula el funcionamiento de una calculadora, con la mayoría de sus características.
Para pulsar las teclas hay que moverse con el cursor hasta la tecla en cuestión y pulsar RETURN.
El funcionamiento es similar al de cualquier calculadora normal:

Tecla	Cómo opera	
sin	número (sin)	.Da el valor del seno (en grados) del número.
cos	número (cos)	.Igual que la anterior.
tan	número (tan)	.Igual que la anterior.
arc	número (arc)	.Usado antes de una de las teclas de funciones trigonométricas da el valor arco seno, coseno...
sto	número (Sto)	.Almacena el número en la memoria.
rcl	(rcl)	.Muestra el contenido de la memoria.
sum	número (sum)	.Suma el número a la memoria.
log	número (log)	.Halla el logaritmo decimal del número.
%	número (x) %	(%). Halla el % del número indicado.
sqr	número (sqr)	.Halla la raíz cuadrada del número.
cl	(cl)	Borra el último número (para los errores).
ac	(ac)	Pone a cero todo (menos la memoria).

Las palabras entre paréntesis son teclas que hay que pulsar.
Las demás teclas son conocidas de todos.
El programa detecta cualquier tipo de fallos, como raíces negativas, divisiones por cero, etc.
Sólo puede pararse por error en el caso de que alguna operación se salga de los límites de 10↑37 ó 10↑-38.
En principio está diseñado para funcionar en un VIC 20, pero puede adaptarse fácilmente a un 64 cambiando las líneas que aparecen a continuación del listado.

```
0 REM VIC CALC POR:ALVARO IBANEZ
1 GOTO40
2 REM SUB DISPLAY
3 PRINTS$"[2CRSRR]"SP$;:IFERTHENER=0
:W=1:PRINT"[5CRSRL][SHIFTE]":RETURN
4 PRINT:S=SGN(A):A=ABS(A)
5 P$=MID$(STR$(A),2)
6 IFS=-1THENP$="-"+P$
7 LP=LEN(P$)
8 IFA>LOR(A<LFANDA<>0)THENP$=LEFT$(P
$,LP-4)+"[SPC]"+RIGHT$(P$,2):LP=LEN(
P$)+3*(A<LFANDA<>0)
9 PRINTS$SPC(17-3*(A>L)-LP)P$;:IFA<L
FANDA<>0THENPRINT"[3CRSRL]-"
10 GOSUB29:A=A*S:DI=A:RETURN
11 I=I+1:IFI<10THENRETURN
12 I=0:PL=SC+Y*22+X
13 IFPEEK(PL)>128THEN16
14 POKEPL,C1+128:IFC2<>32THENPOKEPL+
1,C2+128:IFY<12THENPOKEPL+2,C3+128
15 RETURN
16 POKEPL,C1:IFC2<>32THENPOKEPL+1,C2
:IFY<12THENPOKEPL+2,C3
17 IFA$="[CRSRU]"THENIFY=9THENX=X+2*
(X=14)-(X=6)+(X=18):IFX=10THENX=12:R
ETURN
18 IFA$="[CRSRD]"THENIFY=12THENX=X+(
X=7)-2*(X=12)-(X=17)
19 RETURN
20 IFOP=0THENF1=DI:GOTO27
21 ONOPGOTO22,23,24,25
22 F1=F1+DI:GOTO27
23 F1=F1-DI:GOTO27
24 F1=F1*DI:GOTO27
25 IFDI=0THENER=1:RETURN
26 F1=F1/DI
27 A=F1
28 RETURN
29 IFZE=0THENRETURN
30 PRINTS$SPC(17-LP-ZE+(SN=0))P$;
31 IFSN=0THENPRINT".";
32 IFZE>8THENZE=8
33 PRINTLEFT$(Z$,ZE)
34 RETURN
40 REM DIM ARRAYS
41 DIMA$(6,5)
50 REM STRINGS
51 D1$="[5CRSRL][CRSRD]
52 U1$="[4CRSRL][CRSRU]
53 D2$="[4CRSRL][CRSRD]
54 U2$="[3CRSRL][CRSRU]
55 S$="[HOM][3CRSRD]
56 SP$="[3CRSRR][15SPC]"
57 Z$="000000000"
59 GOSUB5000
60 REM VARIABLES NUMERICAS
61 L=1E9-1
62 SC=7680
63 X=2:Y=21
64 C1=1:C2=3
65 X1=1:Y1=6
66 G=π/180
67 DEFFNS(X)=SIN(X*G)
68 DEFFNC(X)=COS(X*G)
69 DEFFNT(X)=TAN(X*G)
70 DEFFNAS(X)=(ATN(X/SQR(1-X↑2))/G)
71 DEFFNAC(X)=((-ATN(X/SQR(1-X↑2))+(
π/2))/G)
72 DEFFNAT(X)=ATN(X)/G
73 LF=0.01
80 REM COMIENZO
81 POKE36869,242
82 POKE36879,25
83 PRINT"[CLR]
84 GOSUB6010
85 REM ZONA ALTA DEL TECLADO
89 I=-1
90 READA$
91 IFA$="DOWN"THEN99
92 I=I+1
93 IFINT(I/4)=I/4THENPRINT:PRINT:PRI
NT:PRINT"[CRSRR]";
95 GOSUB6000
96 PRINTA$;"[CRSRR][CRSRU]";
97 GOTO90
```

```
98 REM ZONA BAJA DEL TECLADO
99 I=-1
100 READA$
101 IFA$="END"THEN109
102 I=I+1
103 IFINT(I/5)=I/5THENPRINT:PRINT:PR
INT:PRINT"[CRSRR]";
104 GOSUB6005
105 PRINTA$;"[CRSRR][CRSRU]";
106 GOTO100
109 A=0:GOSUB3
110 GETA$:IFA$=""THENGOSUB11:GOTO110
111 IN=-4*(Y>11)-5*(Y<11)
112 IFA$=CHR$(13)THEN150
113 IFA$="[CRSRR]"ORA$="[CRSRL]"ORA$
="[CRSRD]"ORA$="[CRSRU]"THEN115
114 GOSUB11:GOTO110
115 POKEPL,C1:POKEPL+1,C2:IFY<12THEN
POKEPL+2,C3
116 X=X+(A$="[CRSRL]")*IN-(A$="[CRSRR]
")*IN
117 Y=Y+(A$="[CRSRU]")*3-(A$="[CRSRD]
")*3:GOSUB17
118 IFX<0THENX=18+(Y<12)
119 IFX>18THENX=2
120 IFY>21THENY=21
121 IFY<6THENY=6
122 PL=SC+Y*22+X
123 C1=PEEK(PL)
124 C2=PEEK(PL+1)
125 IFY<12THENC3=PEEK(PL+2)
126 I=10:GOSUB11
127 X1=INT((X-2)/4)+1
128 Y1=INT((Y-6)/3)+1
129 GOTO110
150 A$=A$(Y1,X1):IFW=1ANDA$<>"AC"THE
N110
151 IFA$="AC"THEN400
152 IFA$="CL"THEN410
153 IFA$="0"ORVAL(A$)THEN200
154 IFA$="ARC"THEN210
155 IFA$="STO"THEN220
156 IFA$="RCL"THEN230
157 IFA$="SUM"THEN240
158 IFA$="LOG"THEN250
159 IFA$="TPC"THEN260
160 IFA$="SQR"THEN270
161 IFA$="."THEN280
162 IFA$="="THEN290
163 IFA$="SIN"THEN350
164 IFA$="COS"THEN360
165 IFA$="TAN"THEN370
166 IFA$="AD"THEN300
167 IFA$="SU"THEN310
168 IFA$="MU"THEN320
169 IFA$="DI"THEN330
198 W=0:GOSUB3
199 GOTO110
200 REM NUMERO
201 IFNT=1THEN203
202 NT=1:DI=0:PO=0:IFOP=0THENF1=0
203 IFPOTHENPO=PO*10:A=DI+VAL(A$)/PO
:GOTO205
204 A=DI*10+VAL(A$)
205 IFPO<>0ANDA$="0"THEN208
206 ZE=0:IFPOTHENSN=1
207 GOTO198
208 ZE=ZE+1
209 GOTO198
210 REM TECLA ARC
211 AR=1
219 GOTO198
220 REM TECLA STO
221 M=DI:NT=0
222 IFM=0THENPRINT"[HOM][3CRSRD][3CRSRR]
[SPC]":GOTO229
223 PRINT"[HOM][3CRSRD][3CRSRR][SHIFTM]
"
229 A=M:GOTO198
230 REM TECLA RCL
231 A=M:NT=0
239 GOTO198
240 REM TECLA SUM
241 M=M+DI
242 NT=0
249 GOTO198
250 REM TECLA LOG
251 IFDI<=0THENER=1:GOTO198
252 A=LOG(DI)/LOG(10)
259 NT=0:GOTO198
260 REM TECLA %
263 IFOP<>3THEN198
264 TP=F1/100:A=TP*DI
265 OP=0:NT=0:GOTO198
270 REM TECLA SQR
271 IFDI<0THENER=1:GOTO198
272 A=SQR(DI):NT=0
279 GOTO198
280 REM TECLA .
281 IFPOTHEN198
282 PO=1
289 GOTO198
290 REM TECLA =
291 GOSUB20
292 NT=0:OP=0
299 GOTO198
300 REM TECLA +
301 IFOP=1THENOP=0
302 GOSUB20
308 NT=0:OP=1
309 GOTO198
310 REM TECLA -
311 IFOP=2THENOP=0
312 GOSUB20
318 NT=0:OP=2
319 GOTO198
320 REM TECLA X (POR)
321 IFOP=3THENOP=0
322 GOSUB20
328 NT=0:OP=3:DI=0
329 GOTO198
330 REM TECLA / (DIV)
331 GOSUB20
332 IFOP=4THENOP=0
338 NT=0:OP=4:DI=0
339 GOTO198
350 REM TECLA SIN
351 NT=0:IFARTHEN354
352 A=FNS(DI)
353 GOTO198
354 AR=0:IFSTR$(ABS(DI))>="[SPC]1"TH
ENER=1:GOTO198
355 A=FNAS(DI)
359 GOTO198
360 REM TECLA COS
361 NT=0:IFARTHEN364
362 A=FNC(DI)
363 GOTO198
364 AR=0:IFSTR$(ABS(DI))>="[SPC]1"TH
ENER=1:GOTO198
365 A=FNAC(DI)
369 GOTO198
370 REM TECLA TAN
371 NT=0:IFARTHEN376
372 B=DI-INT(DI/360)*360
373 IFB=90ORB=270THENER=1:GOTO198
374 A=FNT(DI)
375 GOTO198
376 AR=0:A=FNAT(DI)
379 GOTO198
400 REM TECLA AC (BORRAR)
401 F1=0:OP=0:ER=0
402 ZE=0:SN=0:AR=0
403 A=0:PO=0:GOTO198
410 REM TECLA CL (ERROR)
411 A=0:PO=0:GOTO198
998 PRINT"[HOM]";:END
999 REM DATA ARRAYS
1000 DATAARC,SIN,COS,TAN,
```

```
1001 DATASTO,RCL,SUM,LOG,
1002 DATATPC,7,8,9,DI
1003 DATASQR,4,5,6,MU
1004 DATACL,1,2,3,SU
1005 DATAAC,0,.,=,AD
1006 REM DATA TECLADO
1007 DATAARC,SIN,COS,TAN
1008 DATASTO,RCL,SUM,LOG
1009 DATA DOWN
1010 DATA"%[SPC]","7[SPC]","8[SPC]",
"9[SPC]",":[SPC]"
1011 DATA"[SHIFT@][COMMT]","4[SPC]",
"5[SPC]","6[SPC]","X"
1012 DATACL,"1[SPC]","2[SPC]","3[SPC]
","-[SPC]"
1013 DATAAC,"0[SPC]",".[SPC]","=[SPC]
","+"
1014 DATA END
5000 REM LEER ARRAYS
5001 FORI=1TO6
5002 FORJ=1TO5
5003 READA$(I,J)
5004 NEXT:NEXT
5996 RETURN
5997 STOP
5998 REM SUBRUTINAS
5999 REM DIBUJA ZONA ALTA DEL TECLAD
O
6000 PRINT"[COMMA][3SHIFT*][COMMF]";
D1$;
6001 PRINT"[SHIFT-][3SPC][COMMK]";D1
$;
6002 PRINT"[COMMC][RVSON][3COMMI][RVSOFF]
[COMMV]";U1$;
6003 RETURN
6004 REM DIBUJA ZONA BAJA DEL TECLAD
O
6005 PRINT"[COMMA][2SHIFT*][COMMF]";
D2$;
6006 PRINT"[SHIFT-][2SPC][COMMK]";D2
$;
6007 PRINT"[COMMC][RVSON][2COMMI][RVSOFF]
[COMMV]";U2$;
6008 RETURN
6009 REM DIBUJA DISPLAY
6010 PRINT"[HOM][2SPC][COMMN][RVSON]
[SHIFTV]IC[SPC][SHIFTC]ALC[RVSOFF][COMMH]
[2SPC][SPC]([SHIFTC])AIB"
6011 PRINT"[SPC][COMMA][18SHIFT*][COMMS]

6012 PRINT"[SPC][SHIFT-][18SPC][SHIFT-]

6013 PRINT"[SPC][SHIFT-][18SPC][SHIFT-]

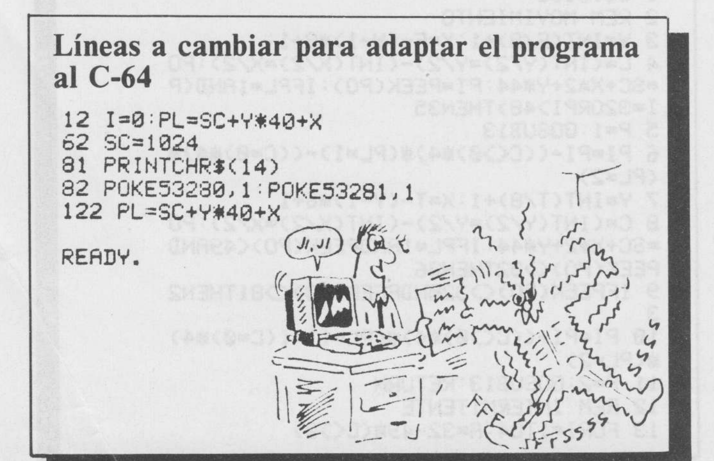
6014 PRINT"[SPC][COMMZ][18SHIFT*][COMMX]

6015 PRINT"[HOM][CRSRD]
6016 RETURN
```

Líneas a cambiar para adaptar el programa al C-64

```
12 I=0:PL=SC+Y*40+X
62 SC=1024
81 PRINTCHR$(14)
82 POKE53280,1:POKE53281,1
122 PL=SC+Y*40+X

READY.
```



Ajedrez (con un reto)

Run. EE.UU.

VIC-20 +3K o superexpander



Este programa está diseñado para funcionar en un Vic-20 con superexpander, aunque también puede funcionar con un cartucho de ampliación de 3K o con uno de 8K-16K si se hacen las oportunas modificaciones.

El programa

Consta de dos partes. La primera se encarga de generar los gráficos y la segunda es el juego propiamente dicho.

Lo que hay que hacer es copiar la primera parte, guardarla en el cassette y comprobar que los gráficos son correctos.

Hecho esto hay que copiar la segunda parte, GUARDARLA antes de hacer RUN (por si los pokes) y repasarla hasta que funcione correctamente.

El juego

El juego en sí es sencillo, porque casi todo el mundo sabe jugar al ajedrez. Si no sabes, este programa puede serte muy útil para aprender las reglas.

Nosotros jugamos con las fichas blancas (en la parte inferior de la pantalla) y el ordenador con las negras.

Para realizar un movimiento (nosotros somos los primeros en jugar) hay que indicar la casilla desde la que movemos y la casilla a la que vamos. Esto se hace mediante un sistema de coordenadas que aparece en todo momento en la pantalla.

Hay que introducir primero la fila (A-H) y después la columna (1-8). No hace falta dejar separación ni pulsar return cuando se acabe.

Una vez hecho esto, nuestra pieza cambiará de lugar, y será el turno del ordenador.

Si se enciende un cuadradito en la parte superior izquierda de la pantalla, quiere decir que el ordenador está "pensando". Cuando acabe, hará su jugada y volverá a ser nuestro turno.

Fin del juego

El juego acaba cuando te comes al rey negro (el Vic no sabe cuándo le estás dando jaque, es un fallo del programa) o cuando el Vic se come tu rey; lo cual es también bastante improbable.

El programa en sí no juega nada bien. Sólo conoce las reglas y por eso mueve sus piezas al azar. Sabe cuándo puede comer, pero no se da cuenta de que le pueden estar amenazando o que podría ganar a la siguiente jugada.

Si intentas hacer alguna jugada ilegal, el ordenador lo advertirá, aunque no siempre.

Como podéis ver, este programa no es ninguna maravilla, pero la intención es haceros ver cómo programar un juego en el que el ordenador juegue contra nosotros usando su inteligencia y ateniéndose a unas reglas.

Este programa puede serviros de ejemplo para el ***CONCURSO DE AJEDREZ COMMODORE WORLD.***

Mandadnos programas que jueguen al ajedrez tal y como hace éste, ¡pero mejor! (que para malo ya está éste). Pueden ser para el VIC-20 o para el C-64 y pueden estar escritos tanto en Basic como en Código máquina.

Eso sí, los programas han de ser vuestros y no copiados de otras revistas o de programas comerciales.

Tenéis que enviarlos antes del 31 de marzo, a:

COMMODORE WORLD
C/ Barquillo, 21. 3º izq.
28004 Madrid

Habrá premio especial para el mejor programa, y saldrá publicado en nuestra revista al mes siguiente.

Listado 1

```
0 REM GENERADOR DE GRAFICOS DEL VIC
CHESS
1 X=-2
9001 C=0:GOTO9003
9002 READC:IFC=0THEN9004
9003 FORJ=0TO7:READK:POKE6144+C*8+J,
K:NEXT:GOTO9002
9004 RESTORE
9006 C=0:GOTO9008
9007 READC:IFC=0THEN9009
9008 FORJ=0TO7:READK:POKE6144+(C+49)
*8+J,255-K:NEXT:GOTO9007
9009 POKE56,23:POKE52,23
9010 FORI=7168TO7679:POKEI,PEEK(I+25
600):NEXT
9011 FORI=7424TO7431:POKEI,255:NEXT
9012 READC:ROM=32768+C*8:X=X+2:IFC=0
THEN9015
9013 FORI=6928+X*8TO6943+X*8STEP2:PO
KEI,PEEK(ROM)
9014 POKEI+1,PEEK(ROM):ROM=ROM+1:NEX
T:GOTO9012
9015 POKE631,76:POKE632,79:POKE633,6
5:POKE634,68:POKE635,13
9016 POKE636,82:POKE637,85:POKE638,7
8:POKE639,13:POKE198,9
9017 END
10000 DATA0,1,2,4,4,30,16,30
10001 DATA1,0,128,64,32,32,120,8,120

10002 DATA2,2,2,2,2,6,12,56,32
10003 DATA3,64,64,64,64,96,48,28,4
10004 DATA4,255,255,254,252,252,254,
240,254
10005 DATA5,255,255,127,63,63,127,15
,127
10006 DATA6,254,254,254,254,254,252,
248,224
10007 DATA7,127,127,127,127,127,63,3
1,7
10008 DATA8,0,0,115,76,76,64,64,64
10009 DATA9,0,0,206,50,50,2,2,2
10010 DATA10,112,24,8,8,56,96,64,64
10011 DATA11,14,24,16,16,28,6,2,2
10012 DATA12,255,255,255,204,204,192
,192,192
10013 DATA13,255,255,255,51,51,3,3,3

10014 DATA14,240,248,248,248,248,240
,192,192
10015 DATA15,15,31,31,31,31,7,3,3
10016 DATA16,14,24,16,16,16,16,24,12
10017 DATA17,192,96,48,156,4,4,4,60
10018 DATA18,4,4,6,2,6,12,56,32
10019 DATA19,48,24,8,56,48,24,14,2
10020 DATA20,254,248,240,240,240,240
,248,252
10021 DATA21,255,127,63,159,7,7,7,63

10022 DATA22,252,252,254,254,254,252
,248,224
10023 DATA23,63,31,15,63,63,31,15,3
10024 DATA24,3,6,12,8,8,8,12,6
10025 DATA25,192,96,48,80,144,16,48,
96
10026 DATA26,4,8,8,4,6,28,48,32
10027 DATA27,32,16,16,32,96,56,12,4
10028 DATA28,255,254,252,248,248,248
,252,254
10029 DATA29,255,127,63,95,159,31,63
,127
10030 DATA30,252,248,248,252,254,252
,240,224
10031 DATA31,63,31,31,63,127,63,15,7

10032 DATA32,0,0,0,0,0,0,0,0
10033 DATA33,1,62,42,32,16,8,4,8
10034 DATA34,128,124,84,4,8,16,32,16

10035 DATA35,8,4,4,4,4,12,56,32
10036 DATA36,16,32,32,32,32,48,28,4
10037 DATA37,255,254,234,224,240,248
,252,248
10038 DATA38,255,127,87,7,15,31,63,3
1
10039 DATA39,248,252,252,252,252,252
,248,224
10040 DATA40,31,63,63,63,63,63,31,7
10041 DATA41,2,14,8,8,14,12,8,8
10042 DATA42,64,112,16,16,112,48,16,
16
10043 DATA43,4,4,4,4,12,56,32,32
10044 DATA44,32,32,32,32,48,28,4,4
10045 DATA45,254,254,248,248,254,252
,248,248
10046 DATA46,127,127,31,31,127,63,31
,31
10047 DATA47,252,252,252,252,252,248
,224,224
10048 DATA48,63,63,63,63,63,31,7,7
10049 DATA110,28,34,93,81,81,93,34,2
8
10050 DATA0
10051 DATA22,9,3,8,5,19
10999 DATA0
```

Listado 2

```
0 REM VIC CHESS.POR:ALVARO IBANEZ
1 GOTO100
2 REM MOVIMIENTO
3 Y=INT(F/8)+1:X=F-(Y-1)*8+1:
4 C=(INT(Y/2)=Y/2)-(INT(X/2)=X/2):PO
=SC+X*2+Y*44:PI=PEEK(PO):IFPL=1AND(P
I=32ORPI>48)THEN35
5 P=1:GOSUB13
6 PI=PI-((C<>0)*4)*(PL=1)-((C=0)*4)*
(PL=2)
7 Y=INT(T/8)+1:X=T-(Y-1)*8+1
8 C=(INT(Y/2)=Y/2)-(INT(X/2)=X/2):PO
=SC+X*2+Y*44:IFPL=1ANDPEEK(PO)<49AND
PEEK(PO)<>32THEN36
9 IFPEEK(PO)<>32ANDPEEK(PO)<>81THEN2
3
10 PI=PI+((C<>0)*4)*(PL=1)+((C=0)*4)
*(PL=2)
11 P=2:GOSUB13:RETURN
12 REM INTERMITENTE
13 FORI=1TO4:A=32-49*(C<>0)
```

```
14 IFP=2THEN16
15 POKEPO,PI:POKEPO+1,PI+1:POKEPO+22
,PI+2:POKEPO+23,PI+3:GOTO17
16 POKEPO,A:POKEPO+1,A:POKEPO+22,A:P
OKEPO+23,A
17 FORJ=1TO150:NEXT
18 IFP=2THEN20
19 POKEPO,A:POKEPO+1,A:POKEPO+22,A:P
OKEPO+23,A:GOTO21
20 POKEPO,PI:POKEPO+1,PI+1:POKEPO+22
,PI+2:POKEPO+23,PI+3
21 FORJ=1TO150:NEXT
22 NEXT:RETURN
23 IFPL=2ANDPEEK(PO)<>41ANDPEEK(PO)<
>45THEN10
24 IFQ=TTHENQ=-1
25 FORI=0TO7:P(I)=P(I)+P(I)*(P(I)=T)
:NEXT
26 IFB(0)=TTHENB(0)=-1
27 IFB(1)=TTHENB(1)=-1
28 IFN(0)=TTHENN(0)=-1
29 IFN(1)=TTHENN(1)=-1
30 IFR(0)=TTHENR(0)=-1
31 IFR(1)=TTHENR(1)=-1
32 IFK=TTHENP=2:GOSUB13:PRINTY$"[RVSOFF]
[5SPC][RVSON]GANASTE":WAIT653,1:RUN
33 IFPL=2THENP=2:GOSUB13:PRINTY$"[RVSOFF]
[7SPC][RVSON]GANE!!!":WAIT653,1:RUN
34 GOTO10
35 E=1:GOTO37
36 T=F:E=2
37 PRINTY$"[RVSOFF][2SPC][RVSON]MOVI
MIENTO[CRSRR]ILEGAL":FORJ=1TO1500:NE
XT:PRINTY$"[RVSOFF][2CRSRR][17SPC]"
38 ONEGOTO130,6
39 REM MOVEMENT COMP.
40 A$=LEFT$(D$,1):B$=RIGHT$(D$,1)
41 DY=(ASC(B$)-67)*Z:DX=(ASC(A$)-67)
*Z:N=0
42 IFX+DX<1ORX+DX>8ORY+DY<1ORY+DY>8T
HENN=1:RETURN
43 PI=PEEK(PO+DX*2+DY*44):IFPI=32ORP
I=81THENRETURN
44 N=1-(PI<49):RETURN
100 GOSUB7000
105 POKE36869,254
110 POKE36879,25
111 POKE36864,14
115 GOSUB8000
125 REM MOV.JUGADOR
130 PRINTY$"[RVSON]?[CRSRR][SPC][CRSRL]
";:CH=0:W=0
131 M$="":FORI=0TO3
132 GETA$:IFA$=""THEN140
133 IFIAND1THEN136
134 IFA$<"A"ORA$>"H"THEN132
135 M$=M$+A$:PRINT"[RVSON]"A$;:GOTO1
38
136 IFA$<"1"ORA$>"8"THEN132
137 M$=M$+A$:PRINT"[RVSON]"A$;
138 IFI=1THENPRINT"[RVSON]-";
139 W=13:NEXT:GOTO147
140 W=W+1:IFW<14THEN132
141 W=0:IFCH=1THEN143
142 PRINT"[RVSOFF][SPC][CRSRL]";:CH=
1:GOTO132
143 PRINT"[RVSON][SPC][CRSRL]";:CH=0
:GOTO132
147 A$=LEFT$(M$,2)
148 B$=RIGHT$(M$,2)
150 F=ASC(LEFT$(A$,1))-65+(8-VAL(RIG
HT$(A$,1)))*8
160 T=ASC(LEFT$(B$,1))-65+(8-VAL(RIG
HT$(B$,1)))*8
165 IFF=TTHENE=1:GOTO37
170 E=0:PL=1:GOSUB2:IFETHEN130
200 REM EL VIC MUEVE
205 PRINTY$;"[RVSOFF][8SPC]"
210 PRINT"[HOM][2CRSRD][19CRSRR][RVSON]
[SPC]";:PL=2
220 E=0:S=RND(1)*6+1:ONSGOSUB300,400
,500,600,700,800
225 IFETHEN220
230 IFS>2THENPRINT"[CRSRL][RVSOFF][SPC]
":GOSUB2
290 GOTO130
300 REM PEON
310 FORI=0TO7:IFP(I)=0THENNEXT:E=1:R
ETURN
320 A=RND(1)*8:IFP(A)=0THEN320
330 Y=INT(P(A)/8)+1:X=P(A)-(Y-1)*8+1
:PO=SC+X*2+Y*44
340 IFPEEK(PO+42)<49ANDPEEK(PO+42)<>
32ANDRND(1)>.2THEND=7:GOTO380
345 IFPEEK(PO+46)<49ANDPEEK(PO+46)<>
32ANDRND(1)>.2THEND=9:GOTO380
347 IFPEEK(PO+44)<>32ANDPEEK(PO+44)<
>81THENE=1:RETURN
350 IFY=2ANDPEEK(PO+88)<>32ANDPEEK(P
O+88)<>81THEN370
360 IFY=2ANDRND(1)>.4THEND=16:GOTO38
0
370 D=8
380 F=X+(Y-1)*8-1:T=F+D:P(A)=T:PRINT
"[CRSRL][RVSOFF][SPC]":GOSUB2:IFY=8T
HEN385
384 RETURN
400 REM REY
405 Y=INT(K/8)+1:X=K-(Y-1)*8+1:PO=SC
+X*2+Y*44:Z=1
410 FORI=0TO7:D$=DK$(I):GOSUB40:IFN=
1THENNEXT:E=1:RETURN
415 A=RND(1)*8:D$=DK$(A):GOSUB40:IFN
=1THEN415
420 F=K:T=K+DX+DY*8:K=T:RETURN
500 REM DAMA
510 IFQ=-1THENE=1:RETURN
520 Y=INT(Q/8)+1:X=Q-(Y-1)*8+1:PO=SC
+X*2+Y*44:Z=1
525 FORI=0TO7:D$=DK$(I):GOSUB40:IFN=
1THENNEXT:E=1:RETURN
530 A=RND(1)*8:D$=DK$(A):GOSUB40:IFN
=1THEN530
540 FORZ=1TO7:GOSUB40:IFN=0THENNEXT
550 Z=Z+(N=1):Z=INT(RND(1)*Z)+1
560 GOSUB40:F=Q:T=Q+DX+DY*8:Q=T:RETU
RN
600 REM ALFIL
610 IFB(0)=-1ANDB(1)=-1THENE=1:RETUR
N
615 J=RND(1)*2:IFB(J)=-1THEN615
620 Y=INT(B(J)/8)+1:X=B(J)-(Y-1)*8+1
:PO=SC+X*2+Y*44:Z=1
625 FORI=0TO3:D$=DB$(I):GOSUB40:IFN=
1THENNEXT:E=1:RETURN
630 A=RND(1)*4:D$=DB$(A):GOSUB40:IFN
=1THEN630
640 FORZ=1TO7:GOSUB40:IFN=0THENNEXT
650 Z=Z+(N=1):Z=INT(RND(1)*Z)+1
660 GOSUB40:F=B(J):T=B(J)+DX+DY*8:B(
J)=T:RETURN
700 REM CABALLO
710 IFN(0)=-1ANDN(1)=-1THENE=1:RETUR
N
715 J=RND(1)*2:IFN(J)=-1THEN715
720 Y=INT(N(J)/8)+1:X=N(J)-(Y-1)*8+1
:PO=SC+X*2+Y*44:Z=1
725 FORI=0TO7:D$=DN$(I):GOSUB40:IFN=
1THENNEXT:E=1:RETURN
730 A=RND(1)*8:D$=DN$(A):GOSUB40:IFN
=1THEN730
735 F=N(J):T=N(J)+DX+DY*8:N(J)=T:RET
URN
800 REM TORRE
810 IFR(0)=-1ANDR(1)=-1THENE=1:RETUR
N
815 J=RND(1)*2:IFR(J)=-1THEN815
820 Y=INT(R(J)/8)+1:X=R(J)-(Y-1)*8+1
:PO=SC+X*2+Y*44:Z=1
```

```
825 FORI=0TO3:D$=DR$(I):GOSUB40:IFN=
1THENNEXT:E=1:RETURN
830 A=RND(1)*4:D$=DR$(A):GOSUB40:IFN
=1THEN830
840 FORZ=1TO7:GOSUB40:IFN=0THENNEXT
850 Z=Z+(N=1):Z=INT(RND(1)*Z)+1
860 GOSUB40:F=R(J):T=R(J)+DX+DY*8:R(
J)=T:RETURN
7000 REM VARIABLES
7001 SC=7702:CO=30720
7002 DIMP(7),B(1),N(1),R(1),DK$(7),D
B$(3),DN$(7),DR$(3)
7003 FORI=0TO7:P(I)=I+8:NEXT
7004 K=4:Q=3
7005 B(0)=2:B(1)=5
7006 N(0)=1:N(1)=6
7007 R(0)=0:R(1)=7
7010 FORI=0TO7:READDK$(I):NEXT
7011 FORI=0TO3:READDB$(I):NEXT
7012 FORI=0TO7:READDN$(I):NEXT
7013 FORI=0TO3:READDR$(I):NEXT
7014 Y$="[HOM][21CRSRD]"
7015 RETURN
7900 DATA BB,CB,DB,BC,DC,BD,CD,DD
7901 DATA BB,DB,BD,DD
7902 DATABA,DA,AB,EB,AD,ED,BE,DE
7903 DATACB,BC,DC,CD
7999 DIBUJA TABLERO
8000 PRINT"[CLR][3SPC][COMMI][COMM@]
[COMM+][SPC][COMM+][COMM£][COMMN][2COMMD]
[13SPC][COMMT][COMMG][COMMM][SPC][COMMM]
[SHIFT£][COMMQ][2COMMZ][SPC][COMMS][RVSON]
AIB[CRSRD]"
8001 PRINT"[2CRSRR]=>[SHIFTA][SHIFTB]
[SHIFTM][SHIFTN][SHIFTR][SHIFTS]π[COMM*]
[SHIFTI][SHIFTJ][SHIFTE][SHIFTF]9:
8002 PRINT"[2CRSRR]?[SHIFT*][SHIFTC]
[SHIFTD][SHIFTO][SHIFTP][SHIFTT][SHIFTU]
[SPC][COMMK][SHIFTK][SHIFTL][SHIFTG]
[SHIFTH];<
8003 PRINT"[2CRSRR]1256125612561256
8004 PRINT"[2CRSRR]3478347834783478
8006 FORI=1TO2:PRINT"[2CRSRR][2SPC][2SHIFTQ]
[2SPC][2SHIFTQ][2SPC][2SHIFTQ][2SPC]
[2SHIFTQ]
8007 PRINT"[2CRSRR][2SPC][2SHIFTQ][2SPC]
[2SHIFTQ][2SPC][2SHIFTQ][2SPC][2SHIFTQ]

8008 PRINT"[2CRSRR][2SHIFTQ][2SPC][2SHIFTQ]
[2SPC][2SHIFTQ][2SPC][2SHIFTQ][2SPC]
"
8009 PRINT"[2CRSRR][2SHIFTQ][2SPC][2SHIFTQ]
[2SPC][2SHIFTQ][2SPC][2SHIFTQ][2SPC]
":NEXT
8010 PRINT"[2CRSRR]@ADE@ADE@ADE@ADE
8011 PRINT"[2CRSRR]BCFGBCFGBCFGBCFG
8012 PRINT"[2CRSRR]LMPQ£]!"CHR$(34)"
-.XYTUHI
8013 PRINT"[2CRSRR]NORS↑←#$/0Z[VWJK
8014 PRINT"[HOM][2CRSRD]"
8015 FORI=8TO1STEP-1
8016 PRINT:PRINT"[18CRSRR][RVSON]"RI
GHT$(STR$(I),1):NEXT
8017 PRINT"[3CRSRR][RVSON]A[CRSRR]B[CRSRR]
C[CRSRR]D[CRSRR]E[CRSRR]F[CRSRR]G[CRSRR]
H"
8018 FORI=7698TO8072STEP22:POKEI+CO,
6:POKEI+1+CO,6:POKEI+2+CO,6:POKEI+3+
CO,6:NEXT
8019 RETURN
```



Búsqueda de errores

Para conocer el contenido de una variable mientras se está ejecutando un programa, basta con pulsar STOP (¡no STOP/RESTORE!), escribir PRINT y el nombre de la variable y a continuación CONT.

M.B. RUN EE.UU.

Magia con el WEDGE

Usando el programa WEDGE con la unidad de discos, si pulsas la barra de espacios mientras estas listando el directorio, el listado se para. Para continuar, basta con pulsar de nuevo ESPACIO.

RUN EE.UU.

Lenguaje máquina

Si quieres cargar y ejecutar fácilmente un programa en lenguaje máquina desde la unidad de discos, un buen truco es crear antes un pequeño programa en basic de la siguiente manera:

```
10 LOAD"PROG C/M",8,1
20 SYS 49152:REM inicio del programa
```

La línea 20 del programa nunca se ejecuta, porque al acabar de cargar el programa, hará un RUN, y se volverá a ejecutar la línea 10 y así ad infinitum. Para que el programa funcione correctamente hay que hacer:

```
10 IF TEST=0 THEN TEST=1:LOAD"PROG C/M",8,1
20 SYS 49152
```

Como las variables se conservan, la línea 10 sólo se ejecutará la primera vez que se haga funcionar el programa.

E.L. RUN EE.UU.

Rutina SAVE

Cuando estas modificando un programa y a la vez estás guardando varias versiones del mismo, puedes automáticamente salvar y borrar la copia anterior de la siguiente manera:

```
59999 end
60000 p$="Programa":open15,8,15
60010 print#15,"s0:"+pn$+".bkup"
60020 print#15,"r0:"+pn$+".bkup="+pn$
60030 close15
60040 save pn$,8
```

Cuando quieras salvar la última versión, teclea: RUN 60000. De este modo siempre existe la última copia y la anterior. El nombre del programa no puede superar los 11 caracteres.

J.L. RUN EE.UU.

Errores en los listados

Si alguna vez te ha ocurrido que al listar un programa éste termina repentinamente con un REM seguido de SYNTAX ERROR, puede ser debido a que tras el REM hubiera un SHIFT L oculto. Para solucionarlo sólo tienes que colocarle en la línea del REM y pulsar return.

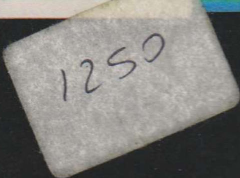
P.J. RUN EE.UU.

Copiar un programa

Cuando vayas a sacar una copia impresa de un programa, usa los códigos ASCII para las funciones del cursor o de borrado de pantalla. Por ejemplo [CLR] es igual que chr$(147). Te costará un bit más, pero ganarás en claridad a la hora de depurar tus problemas.

D.M. RUN EE.UU.

SOFTWARE ESPAÑA



SOFTWARE ESPAÑA
Avenida de Arteijo, 19
15004 La Coruña
Teléf. (981) 25 51 72
Télex 47206 PPLL-E

